

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**



**FEUP**

# **Multi-Criteria Decision Analysis on Content Management System Projects**

**Simão Belchior de Castro**

FINAL VERSION

Report of Project/Dissertation  
Master in Informatics and Computing Engineering

Supervisor: Teresa Galvão Dias (PhD)

29-June-2009



# **Multi-Criteria Decision Analysis on Content Management System Projects**

**Simão Belchior de Castro**

Report of Project/Dissertation  
Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: António Augusto Sousa (Associated Professor at Faculdade de Engenharia da Universidade do Porto)

---

External Examiner: Luís Manuel Borges Gouveia (Associated Professor at Universidade Fernando Pessoa)

Internal Examiner: Teresa Galvão Dias (Auxiliar Professor at Faculdade de Engenharia da Universidade do Porto)

29-June-2009



# Abstract

With the proliferation of Internet access all companies, from small to large ones, that previously were unaware of it perceive the importance of having their own Web site. As a channel of communication, a Web site allows companies to contact with their clients, gather new clients, and display their products/services to the market.

Content Management Systems are the ideal Web applications to provide such channel of communication. With their functionalities of content addition and edition, CMSs are the technologies that can avoid Web sites to become outdated. There are two main requirements to develop a CMS: CMS Tool and a hosting solution, where the application will run and be made available through the Internet. Given the large number and diversity of CMSTs and hosting solutions available, the objectives pursued and restrictions faced by firms, choosing how to do a CMS can be usefully regarded and dealt with as a Multi-Criteria Decision Analysis problem.

At Tam Tam B.V., a Dutch Web Development company, otherwise simply referred as Tam Tam,, CMS projects are currently developed with reference to two different scenarios, using two different CMST and the same hosting solutions.

A conjugation of two very distinct factors has made Tam Tam elaborate on a problem related to their CMS Projects. These factors were the global crisis that is threatening some of Tam Tam's clients, Tam Tam's permanent concern with innovation. The problem was to find alternatives to the way Tam Tam currently develops its CMS Projects. Such alternatives should simultaneously result in a reduction in costs and an increase in innovation. The main objective of the project reported this report addresses was helping the firm solve this problem.

This report describes the steps taken towards the definition of different alternatives for the CMS projects. These alternatives had to be composed of a CMST and a hosting solution. Keeping its goals in mind some constraints were defined by Tam Tam. The analysed CMST should be Open Source, in opposition to Tam Tam's traditional use of proprietary tools, and the hosting solution should be based in a new computing style known as Cloud Computing.

The focus of this project was then to define alternative scenarios for Tam Tam's CMS Projects, which should be characterized and compared. A second large objective was to analyse Multi-Criteria Decision methodologies that could help Tam Tam managers consistently make their decisions. It was also part of the objectives of the project to guide Tam Tam's Decision Makers in using the Multi-Criteria Decision methodology proposed. Finally, a Web application was developed to implement the methodology proposed and adopted, with the objective of simplifying the company's decision processes.



# Resumo

Com a generalização do acesso à internet um grande número de empresas, das pequenas às maiores, começou a aperceber-se da importância de ter o seu próprio site na Internet. A Internet tornou-se um canal de comunicação poderoso e incontornável.

Content Management Systems (CMS) são gestores de conteúdos e ideias Web para gerir o canal de comunicação entre as empresas e os seus clientes. Os CMSs possuem funcionalidades que permitem a fácil adição e edição de diferentes tipos de informação e facilitam extraordinariamente a tarefa de actualização dos sites das empresas, que pode ser feita distribuída por múltiplos responsáveis funcionais a vários níveis,. Para o desenvolvimento de um CMS há dois requisitos principais: uma ferramenta CMS e uma solução de alojamento Web, onde os sistemas resultantes serão instalados e ficarão acessíveis através da Internet. A quantidade de ferramentas CMS e de serviços de alojamento existentes é muito vasta e diversa e, por isso o processo de decisão acerca de como construir um CMS não é trivial e pode, com vantagem, ser visto como um processo de decisão multi-critério.

Actualmente na Tam Tam B. V. (também simplesmente referida como Tam Tam), uma empresa Holandesa de desenvolvimento Web, os projectos de CMS são desenvolvidos com base em dois cenários diferentes.

A conjugação de dois factores distintos levou a que a Tam Tam decidisse debruçar-se com redobrada atenção sobre o problema do modo de desenvolvimento dos seus projectos de CMS. Esses factores foram, a crise económica mundial que tem ameaçado alguns dos seus clientes e o valor que a Tam Tam sempre atribuiu à inovação. O problema definido consistiu em encontrar alternativas aos cenários actualmente seguidos pela Tam Tam para o desenvolvimento de projectos CMS. Essas alternativas deveriam conseguir conjugar uma redução dos custos e uma mudança dos processos comumente utilizados pela Tam Tam. Encontrar soluções para este problema foi o objectivo principal do projecto descrito no presente relatório.

Este relatório descreve os passos seguidos para a definição de um conjunto de alternativas para estes projectos. As alternativas incluíam a escolha de uma ferramenta de desenvolvimento de CMSs, de preferência de código aberto, e de uma alternativa de alojamento Web, havendo especial interesse por alternativas baseadas em Cloud Computing.

Para além da definição das alternativas outro objectivo deste projecto foi ajudar os decisores a escolherem uma metodologia de análise multi-critério que permitisse auxiliar a Tam Tam na tomada destas decisões. Durante o projecto foi desenvolvido um programa que implementou a metodologia escolhida, com o objectivo de tornar o processo de decisão mais simples.





# Acknowledgements

Having followed the course of Informatics Engineering the chances are that I will never gain an Oscar from the Hollywood Academy of Motion Picture Arts and Sciences, and I will not be able to give a wining speech or any speech at all, for that matter. So I would like to use this humble page to utter a word of appreciation to everyone who I know would be there because of me as an actor as otherwise they have been during my life, my studies and during this project.

First I would like to thank Stef van Hooijdonk my supervisor at Tam Tam for all the help and support. Pascal Lagarde project manager at Tam Tam for all the help. And everyone at Tam Tam for so warmly welcoming me to The Netherlands.

I would also like to thank my supervisor from FEUP, Prof Teresa Galvão Dias for her precious help, guidance, and support during my project.

A special thanks goes to Neel from Tam Tam, to Hugo Peixoto from MIEIC, and to my father for helping me with my report, for some proofreading it and for giving some feedback.

And now for much more than just this project I would like to thank to my parents, for everything that they taught me, for being there when I need them, for being an inspiration, for being who they are, for being my parents, for bestowing me their love.

I also want to thank to my sisters, for putting up with me, I know it wasn't always easy!

To my nephews, brothers in law, my *parra* cousins, my godfather, my aunt São, my rugby cousins and all my family, also a thank you for being part of my life.

A big thank you for everything to my “Priminhas”, to the “Ciganos”, to the “Ocupados”, to the “Confraria do Caldeirão”, to the GeekTribe and also to the members of “Culturacto”.

In the beginning of this semester a group of students from MIEIC thought that the boots would be beaten, but we dodged it. A thank you to all the VMT members for the exchange of knowledge, desperation, frustration, motivation and support!

Last but definitely not least I want to thank Mariana for all the support, all the patience for my bad mood, for being an inspiration and a role model of how to do things right. Thank you for being there with me, for everything. Thank you for sharing The Netherlands with me!

Finally I wish to acknowledge, that despite all the help that directly or indirectly contributed to my carrying out this project and submitting the current report, whatever was or is not correct or could be made better in either is entirely my sole responsibility.

Simão Belchior de Castro



# Table of Contents

|   |           |
|---|-----------|
| <b>Chapter 1.....</b>                               | <b>1</b>  |
| Introduction .....                                  | 1         |
| 1.1 Context.....                                    | 1         |
| 1.2 Project.....                                    | 3         |
| 1.3 Motivation and Objectives.....                  | 3         |
| 1.4 Report Overview.....                            | 4         |
| <b>Chapter 2.....</b>                               | <b>5</b>  |
| Content Management Systems and Cloud Computing..... | 5         |
| 2.1 Content Management System .....                 | 5         |
| 2.1.1 Lifecycle.....                                | 7         |
| 2.1.1.1 User Roles.....                             | 7         |
| 2.1.1.2 Lifecycle Phases.....                       | 9         |
| 2.1.2 Content Management System Tools.....          | 11        |
| 2.2 Web Hosting.....                                | 12        |
| 2.3 Cloud Computing.....                            | 14        |
| 2.3.1 The Cloud.....                                | 14        |
| 2.3.2 Cloud Computing .....                         | 17        |
| 2.3.2.1 Advantages.....                             | 18        |
| 2.3.2.2 Risks.....                                  | 19        |
| 2.3.2.3 Future of Cloud Computing .....             | 20        |
| 2.3.3 Cloud Services Providers.....                 | 21        |
| 2.4 Summary.....                                    | 23        |
| <b>Chapter 3.....</b>                               | <b>25</b> |
| Multi-Criteria Decision Analysis.....               | 25        |
| 3.1 Multi-Criteria Decision Analysis.....           | 25        |
| 3.2 Methodologies .....                             | 28        |
| 3.2.1 Multi-Attribute Utility Theory.....           | 28        |
| 3.2.2 AHP .....                                     | 29        |
| 3.2.3 Outranking methods.....                       | 30        |
| 3.3 Summary.....                                    | 31        |

|  |           |
|--|-----------|
| <b>Chapter 4.....</b>  | <b>33</b> |
| Problem Overview and Approach .....  | 33        |
| 4.1 The Main Problem .....   | 33        |
| 4.1.1 Research and Definition .....  | 34        |
| 4.1.2 Characterization and Comparison.....   | 35        |
| 4.2 The Second Problem.....  | 35        |
| 4.3 Summary.....   | 35        |
| <b>Chapter 5.....</b>  | <b>37</b> |
| The Alternative Scenarios .....  | 37        |
| 5.1 Scenarios Definition .....   | 37        |
| 5.1.1 Research.....  | 38        |
| 5.1.2 First Draft .....  | 38        |
| 5.1.3 The Scenarios .....  | 39        |
| 5.2 Scenarios Characterization .....   | 41        |
| 5.2.1 The Criteria.....  | 41        |
| 5.2.2 Characterization.....  | 43        |
| 5.2.2.1 Scenario 1: Microsoft Office SharePoint Server 2007 + Amazon Web Services 43 |           |
| 5.2.2.2 Scenario 2: Custom CMS + Microsoft's Azure Services Platform.....            | 46        |
| 5.2.2.3 Scenario 3: Umbraco + TerreMark .....  | 48        |
| 5.2.3 Scenarios Variations .....   | 50        |
| 5.3 Summary.....   | 53        |
| <b>Chapter 6.....</b>  | <b>55</b> |
| MCDA Choice and Implementation.....  | 55        |
| 6.1 Methodology Selection.....   | 55        |
| 6.2 Implementing the AHP Methodology.....  | 56        |
| 6.2.1 Requirements Specification.....  | 57        |
| 6.2.1.1 Application Perspective .....  | 57        |
| 6.2.1.2 User Interfaces .....  | 57        |
| 6.2.1.3 Software Interfaces.....   | 57        |
| 6.2.1.4 Communication Interfaces .....   | 57        |
| 6.2.1.5 Application Functionalities .....  | 57        |
| 6.2.1.6 User Characteristics .....   | 58        |
| 6.2.1.7 Constraints .....  | 59        |
| 6.2.1.8 Sequence of Use.....   | 59        |
| 6.3 Implementation Details.....  | 60        |
| 6.3.1 Implementation of the AHP Process.....   | 60        |
| 6.3.2 Handling the Information .....   | 65        |
| 6.3.3 Application Overview.....  | 66        |
| 6.3.4 Summary.....   | 67        |
| <b>Chapter 7.....</b>  | <b>69</b> |

|  |           |
|--|-----------|
| MCDA Applied to the Problem.....                       | 69        |
| 7.1 Applying the MCDA Methodology.....                 | 69        |
| 7.1.1 Information Preparation.....                     | 69        |
| 7.1.2 Applying AHP.....                                | 71        |
| 7.1.2.1 Filling-in the Criteria and the Scenarios..... | 71        |
| 7.1.2.2 The Pair Wise Comparisons.....                 | 73        |
| 7.1.2.3 AHP Results.....                               | 74        |
| 7.1.3 Results Analysis.....                            | 76        |
| 7.2 Summary.....                                       | 79        |
| <b>Chapter 8.....</b>                                  | <b>81</b> |
| Conclusions and Future Work.....                       | 81        |
| 8.1 Conclusions.....                                   | 81        |
| 8.2 Future Work.....                                   | 82        |
| <b>References .....</b>                                | <b>85</b> |
| <b>Index .....</b>                                     | <b>93</b> |



# List of Figures

|   |    |
|---|----|
| Figure 1 Content Management System Lifecycle.....   | 9  |
| Figure 2 Youseff et al Cloud Ontology [71] .....  | 16 |
| Figure 3 Amazon Web Services – Medium Server Farm .....   | 44 |
| Figure 4 Web Application Use Cases .....  | 58 |
| Figure 5 Sequence of Use .....  | 59 |
| Figure 6 Page of the first step of the AHP decision aid process – Defining the criteria   | 61 |
| Figure 7 Page of the second step of the AHP decision aid process – Defining the<br>Scenarios .....  | 62 |
| Figure 8 Page of the third step of the AHP decision aiding process- Pair wise<br>comparison of the criteria .....                                   | 63 |
| Figure 9 Page of the fourth step of the AHP decision aiding process- Pair wise<br>comparison of the scenarios in relation to the each criteria..... | 63 |
| Figure 10 Results Page of the AHP decision aiding process .....   | 64 |
| Figure 11 Example of a saved configuration file.....  | 65 |
| Figure 12 Loading a saved configuration .....   | 66 |
| Figure 13 Field Validation (Red fields) .....   | 67 |
| Figure 14 Applying AHP - Criteria List .....  | 72 |
| Figure 15 Applying AHP - Scenarios Information .....  | 73 |
| Figure 16 Applying AHP - Comparing Sub Criteria .....   | 74 |
| Figure 17 Applying AHP - Criteria Ranking.....  | 75 |
| Figure 18 Applying AHP - Results 1 .....  | 75 |
| Figure 19 Applying AHP - Results 2.....   | 76 |
| Figure 20 Applying AHP - Results 3.....   | 76 |
| Figure 21 Applying AHP - Compare Top 3 Scenarios .....  | 77 |
| Figure 22 Applying AHP - Scenario 1 vs Scenario 3 .....   | 78 |





# List of Tables

|                                      |    |
|--------------------------------------|----|
| Table 1 AHP Judgement Scale .....    | 30 |
| Table 2 Criteria Scales .....        | 70 |
| Table 3 Scenarios Information.....   | 70 |
| Table 4 Scenarios Denomination ..... | 73 |



# Abbreviations

|      |                                    |
|------|------------------------------------|
| AHP  | Analytic Hierarchy Process         |
| ANP  | Analytic Network Process           |
| API  | Application Programming Interface  |
| ASP  | Active Server Pages                |
| AWS  | Amazon Web Services                |
| CMS  | Content Management System          |
| CMST | Content Management System tool     |
| EC2  | Elastic Compute Cloud              |
| HTML | Hypertext Markup Language          |
| HTTP | Hypertext Transfer Protocol        |
| IT   | Information Technology             |
| MAUT | Multi-Attribute Utility Theory     |
| MCDA | Multi-Criteria Decision Analysis   |
| MCDM | Multi-Criteria Decision Making     |
| MOSS | Microsoft Office SharePoint Server |
| PHP  | Hypertext Preprocessor             |
| RAM  | Random Access Memory               |
| S3   | Simple Storage Service             |
| SQL  | Structured Query Language          |
| WWW  | World Wide Web                     |



# Chapter 1

## Introduction

*It has to start somewhere It has to start sometime  
What better place than here, what better time than now?*  
Rage Against the Machine – Guerilla Radio

This chapter briefly contextualizes this project, presenting the company where it was developed, the motivations behind it, as well as the objectives it aims to fulfil. In the end of this chapter an overview of the report is presented.

### 1.1 Context

Content Management Systems (CMS) are Web applications that have functionalities that make it easy to create, edit, and publish content on-line. With the proliferation of Internet access Web development in general and this type of systems are becoming more and more popular. Small, medium, and big companies are building or buying their own CMS.

To build a CMS it is necessary to choose one of two paths. Either build one from scratch or use a CMS Tool (CSMT<sup>1</sup>) to build it. CMST are applications that after being installed in a Web server provide a standard CMS with basic functionalities. These tools come with a set of basic functionalities that allow the user to expand and configure it with relative ease. The use of a CMST simplifies the process of building a CMS, reducing the time necessary to do it.

Companies seeking to build their own CMS through the use of a CMST which CMST they will use. The CMS will have to be hosted somewhere to be accessible through a normal Internet connection, and so this is also a concern that companies must have when building a CMS. Both

---

<sup>1</sup> Content Management System is a noun used to refer to different things: the type of web applications known as Content Management Systems; the tools used to create a web application of the Content Management System type. So in this report it will be used the abbreviation CMS for the type of web application and CMSt for the tools that help on the development of this type of web application.

in terms of CMST and of hosting solutions the number of existing alternatives is very large. In both cases the existing alternatives can be somewhat similar or very different.

The existence of so many alternative tools and hosting solutions make the choice of how to build a CMS a Multi-Criteria Decision problem.

Tam Tam B.V. [1] is a full-service internet company based in the Netherlands. Tam Tam was founded in 1996 by two brothers, Bart and Paul Manuel, and since then it has been growing in size and importance in The Netherlands. The company has been profitable since the first year and had a turnover of 8.4 million Euro in 2008[2]. As of May 2009 Tam Tam had over 100 employees and two offices, the headquarter in Rijswijk and a newly opened office in Utrecht.

Tam Tam is specialized in developing an on-line strategy for other companies, doing consultancy, design and implementation of online marketing and communication, corporate portals and business applications, basing all its work in three core values: Service, Innovation and Fun. The main markets of Tam Tam are education, healthcare, government, business to business and finance. Tam Tam is a Microsoft Gold Certified Partner[3] having, as a result, access to the latest technologies from Microsoft and providing its customers with the newest possible solutions. One of Tam Tam's motivations is to put its clients' world on-line.

As of January 2009, Tam Tam is structured in small but empowered advisory groups allowing a good specialization but also a very agile way of working, with high levels of interoperability. These advisory groups have the following structure:

- Content Management Systems;
- Communication Utrecht "Yes we can!";
- Customer Self-Service;
- New Collaboration Group
- Operational Services;
- Portals and Integration;

Together with this structure comes a natural division of projects among the different groups. And if there are companies seeking for an all around solution ranging from intranet to internet systems, there are some others whose needs and requirements are less complex. Companies with a smaller dimension normally just look for a solution for their public Web site, normally a CMS.

In Tam Tam, the Content Management Systems (CMS) Advisory Group is the group that is responsible for the development of CMS projects. For the development of these projects, the CMS group usually uses one of two tools: Microsoft Office SharePoint Server(MOSS)[4] and SmartSite[5], depending on different factors or on the customers preferences. Most of the applications developed are either hosted at Terremark's data center in Amsterdam[6], where Tam Tam has its servers running, or in the company's on-premises servers.

## 1.2 Project

For a company like Tam Tam, whose core values include service and innovation and who dwells in the very competitive field of web development, there are different challenges that it must face each day to stay in the top and to provide the best to its customers.

One of these challenges is related with the ever-changing world of web development, with new tools and/or paradigms emerging every day. The company must be aware of the changes and of the evolution in the market so it can be sure to use the best and only the best.

Another challenge is common to every company in the world with independently of the field of activity. That is the economical challenge. This is ever so true when the world is facing a big and global economical crisis that threatens all companies, no matter their size. This crisis is making companies cut on their budgets, making them less prone to spend money on too complex products.

Even though these challenges are different, they are not totally independent. In the field of computers, the adoption of new technologies and new solutions cannot be done without careful thinking, due to the cost of change and adoption. The advantages of adoption must be weighed against these costs.

Obviously, these challenges exist for all the groups of the company, but this project will focus only in the CMS Advisory Group.

With the popularization and recent breakthroughs in cloud computing and the growing number of CMST Tam Tam wants to analyze and compare different options that might use this new computing style and different types of CMST.

## 1.3 Motivation and Objectives

Having already developed many different projects with MOSS and SmartSite, the CMS group has seen its skills developed. Due to this fact, it is easier for the group to develop new projects that come by. But as it has been said before, Tam Tam strives to bring innovation to all of its projects and CMS projects are no different. And so the will to innovate is one of the motivations behind this project.

Another motivation which is also related with innovation is the will to explore other alternative tools for building CMS projects. Among the different alternatives there is some curiosity about Open Source tools, a growing paradigm of software development. The fact that Tam Tam, being a Microsoft Gold Partner, is willing to look into Open Source solutions, which normally in the common sense are seen as the enemy of Microsoft, shows the costs saving motivation, since Open Source technologies tend to be cheaper than enterprise solutions.

Related to costs saving is also Cloud Computing , “a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet”[9]. This style of computing is becoming one of the trends of IT world in 2009[10, 11], with a rising number of Cloud Services Providers and the development of supporting technologies.

The objectives of this project are the following:

- Research and document the state of the art of Content Management Systems;

- Research and document the state of the art of Cloud Computing and Cloud Service Providers, as well as the available Cloud Services;
- Define and analyze different scenarios of composed of CMST plus a hosting solution to be used in the development of CMS projects at Tam Tam;
- Research and choose a method to help Tam Tam analyze the different scenarios specified;
- Present and guide Tam Tam's Decision Makers through the decision process of choosing the most profitable scenario for the company;
- Present the results of the decision process of choosing the best scenario and analyze that option as well as the ranking of the different options;

## 1.4 Report Overview

The rest of the report is organized as follows:

- **Chapter 2** "Content Management Systems and Web Hosting" introduces the concepts of Content Management Systems and Web Hosting. This chapter focus in the origin of CMS, and also presenting details about this type of applications' lifecycle and most common characteristics. Regarding Web Hosting it details the different types of Web Hosting, and focus in one of those types, Cloud hosting which is also a service provided by Cloud Computing. Following this it presents the concepts behind Cloud Computing, making an overview of this computing style.
- **Chapter 3** "Multi-Criteria Decision Analysis" presents this branch of Operational Research. It also looks into some MCDA methodologies.
- **Chapter 4** "Problem Overview and Approach" explains with further detail the main problem behind this project and the approach that was defined to solve it.
- **Chapter 5** "The Alternative Scenarios" is presented the alternative scenarios definition process. Presenting the scenarios that were defined and also characterizing them. This chapter also introduces the criteria used in the characterization of the scenarios.
- **Chapter 6** "MCDA Choice and Implementation" introduces the choice of the MCDA methodology that was applied to the problem. In this chapter it is also detailed the development of an Web Application that implements the chosen methodology.
- **Chapter 7** "MCDA Applied to the Problem" describes the process of applying the chosen MCDA methodology to the problem. This was done together with Tam Tam's Decision Makers, and through the use of the Developed Web Application.
- **Chapter 8** "Conclusions and Future Work" reviews the project, draws the necessary conclusions, and ends up by pointing out some ideas regarding future work that could be done.



## Chapter 2

# Content Management Systems and Cloud Computing

*You can't manage knowledge – nobody can. What you can do is to manage the environment in which knowledge can be created, discovered, captured, shared, distilled, validated, transferred, adopted, adapted and applied.*

Chris Collison and Geoff Parcell[7]

This starts by building the required base of knowledge concerning Content Management Systems, and Web Hosting. Following the presentation of different Web hosting types a section of this chapter is dedicated to Cloud Computing, as this is the technological base for Cloud hosting. In each section a short technological review is made. This chapter will prepare the reader for the following chapters.

### 2.1 Content Management System

In the field of Information Technology (IT) and in the World Wide Web (WWW) content can take the form of text, multimedia files or any other type of files[8]. As of May 2009 the “World’s digital content [is] equivalent to stack of books stretching from Earth to Pluto 10 times”[9], having reached the total of 487billion gigabytes (GB).

All this content is not created randomly and must be managed and maintained. These tasks are referred to as Content Management (CM) which can be described as the “set of processes and technologies that support the evolutionary life cycle of digital information”[8].

CM has been around ever since WWW infancy, when “Web sites largely consisted of static hand crafted HTML code on convenient text editors”[10]. At that time the maintenance of Web sites was done by a Webmaster[11] or by a team of Webmasters and it was almost as

complex as creating a new website. Since the Web pages were static it was necessary to go deep into the HTML code to add, remove or edit the content.

In the late 1990's new changes took place in the way Web sites were build and their complexity increased in various dimensions: content volume increased, dynamic content started to be used, the number of hits per Web site increased and so did the hardware and software complexity[10]. One of the most important changes is related with the increasing number of visitors. With more people having access to the Internet, the visibility of websites increased and the tolerance to errors decreased[9].

The evolution of Internet made it more interesting to companies but it also forced those companies to review their Web sites. It was not enough anymore to have a simple showcase or information repository Web site. Companies moved to portals, that even worked as a new place of business[9].

With this change in the functions and purpose of the Web sites there was the need to review the way they were maintained and managed. Being places of business it became essential to have ways of having the content updated instantly and constantly. And it has also became important for the people working closely with the different parts of business to have ways of doing the content management, and not only the Webmaster or the team responsible for the creation of the Web site[9].

The companies need tools to help them manage their websites in a more dynamic way. Those tools are Content Management Systems, which can be seen as ways to automate the process of managing content in Web sites.

Even though the noun Content Management System seems to speak for itself there are different views on what really is a CMS. There are those who think that CMS is a general definition for the whole group of technologies that allow for the management of digital content (and so they think that Enterprise Content Management (ECM), Web Content Management (WCM), Document Content Management (DCM), etc. are different types of Content Management Systems, depending on the type of content managed as presented in the Wikipedia page about Content Management System [12]). Others think that there is a big difference between a Content Management System and a Web Content Management, with none of them deriving from the other [13].

Accepting the first point of view as true, the following definition, by ContentManager.EU.Com[14], a company working closely with Content Management Systems, seems to correctly define a CMS: "A CMS is a tool that enables a variety of (...) technical and (...) non technical staff to create, edit, manage and finally publish (...) a variety of content (...), whilst being constrained by a centralized set of rules, process and workflows that ensure coherent, validated electronic content."[15].

Based on this definition, whenever CMS projects are referred in this report it is referred to the general idea of Content Management System, without focusing on the type of content. This is due to the variability of the type of content in the different CMS Projects built by Tam Tam, which are dependent basically on the client's requirements.

There are various characteristics generally associated with a CMS. Some of them can be spotted in the above definition while others can be found in different Web sites[16-19]. In the following we present only those that make the CMS such a strong tool:

- Separation of administration functionalities from visitors and users functionalities;
- Visual interfaces for adding, editing and removing content from the Web site, without needing any knowledge of HTML or any other programming language;
- Existence of user roles to allow different users to access and manage different parts of the content;
- Possibility of different users to edit the content of the Web site at the same time;
- Versioning of the Web site content, supported by the database;
- Separation of the layout from the content, keeping the layout integrity;
- Existence of workflow schemas that make it mandatory to administrators to approve some content before it gets published.

Ever since their dawn CMS have been evolving, allowing for more and more different functionalities that make the task of maintaining an interesting and up to date Web site easier. Still, there is a technical part behind the deployment of a CMS. The Web site must be built and configured by Web developers so that later on the content can be managed by other users. To do this there are different alternatives available. Some companies choose to build their own CMS, others buy a product from other companies, some others buy the a tool that allows them to easily build a CMS Web site and, finally, others use free and Open Source tools and build their own CMS Web site. Due to the context of this project, the approach will be based on the point of view of a company like Tam Tam, which builds Content Management Systems for its customers.

### 2.1.1 Lifecycle

As it will be shown further ahead there are different tools designed to easily develop Content Management Systems. But regardless of the tools used, there are some features common to all CMSs, which are the user roles and the lifecycle of the system. The lifecycle can be divided into two main phases. These two phases will be referred to as the **setup phase** and the **content production phase**. Each phase is managed by different user roles.

Before looking into the different phases of the lifecycle of a CMS one must know and understand which type of users exist and interact with the system throughout its lifecycle.

#### 2.1.1.1 User Roles

A Content Management System can have different user roles. Nevertheless, the roles existing in a CMS that is up and running do not necessarily represent all the intervenient in the different phases of the lifecycle. There are some users whose functions are specifically related with the setup phase and as such they finish when the lifecycle reaches the content production phase. On the other hand, the user roles that exist in the content production phase exist during all the phases of lifecycle even though they usually are not active on the setup phase.

During the two main phases there is one user role, the administrator, who is controlled by different users in each of the main phases. This role is controlled by the development team

during the setup phase and is handed over to the IT responsible of the company that bought the CMS during the transition from the setup phase to the content production phase.

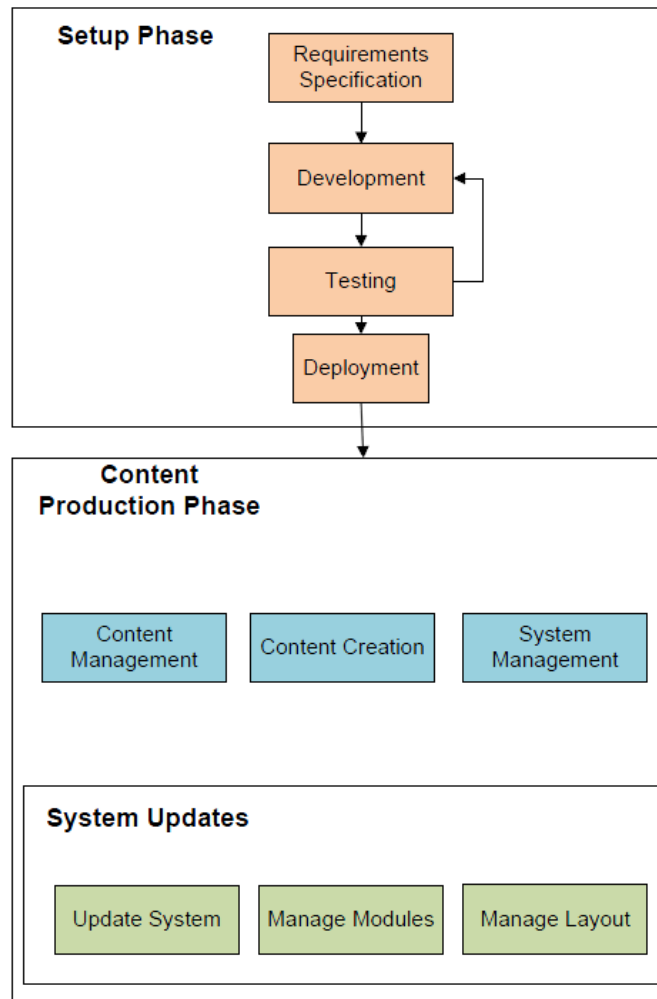
Now, let us take a closer look into the different roles, keeping in mind that in a real life scenario there can be more than one user under a user role:

- **Graphical Designer:** The graphical designer is responsible for designing the graphical layout of the CMS. He is a member of the development team that mainly acts during the setup phase;
- **Functional Designer:** The functional designer is responsible for defining the different modules that should be part of the system, and that normally are defined in a requirements specification document built with the client. He also is a member of the development team;
- **Developer:** The developer is the IT professional who will implement both the layout and the functions defined by the graphical and the functional designer. Basically, he is responsible by putting together both functionality and graphics that satisfy the customer needs. The developer is the last member of the development team;
- **Administrator:** The administrator is the super user of the CMS and is the role of the IT responsible of the company that acquired the system. He is responsible for managing all parts of the system. Some of the most common functions of the administrator are the management of users, content, and content types. Depending on the specifications of the system he might also be able to change the organization and the layout of the CMS without much effort;
- **Content Manager:** The content manager is someone that the company trusts and who is responsible to manage the content of the system. Normally this user is able to create, edit and delete content on the whole system. If some type of workflow feature is defined and if it is necessary to approve content before publishing, it is the content manager who is responsible for the approval of that content. This user role must have enough permissions to grant or deny the access of the other users to specific parts of the system, normally allowing or disallowing content editors to produce content to those parts of the system;
- **Content Editor:** The content editor is normally the main responsible for the content of the Content Management System. This user is the one that creates most of the content of the system and is normally the user that keeps the system up to date. Sometimes the content created by the content editor has to be approved by a content manager;
- **Visitor:** The visitor is any user that reaches the system and that does not have or did not provide his credentials. The visitor is able to view the public content of the system, to visit the different sections and even to interact with it if there are any forms in the system, like for instance a contact form. In the business world, the visitor is normally a client or a possible client of the owner of the CMS and is the target of the system. Any CMS should be built to inform its visitors about the

company's activities.

### 2.1.1.2 Lifecycle Phases

The CMS lifecycle is composed by two main phases as represented in Figure 1. The figure also represents the actions and sub phases that take place during each of the two main phases. The figure also shows that there is a third phase named System Updates that is contained into the content production phase. These three phases will now be presented in detail.



*Figure 1 Content Management System Lifecycle*

- 1- Setup Phase:** The setup phase is the first of the lifecycle of a CMS. In this phase, the system is defined, configured, installed, and deployed. The users that participate and interact directly with the system compose the development team. In the above figure one can see that this phase is constituted by four sub phases: Requirements Specification, Development, Testing and Deployment. This is just one of the many possible divisions and organizations of the setup phase, but it seemed that it would make it easier to explain which are the main steps taken during this phase. In the Requirements Specification, the development team should meet with the client so that

together they can identify the requirements of the system, including functional and non functional requirements. The functional designer will be one of the most active member of the development team. The resulting specification will then help the development team in the next sub phase. Using the information gathered from the client and organized by both the team and the client the team can start the development. Depending on the chosen tool or if it is a CMS made from scratch, the development might vary. If the project consists in the development of a CMS from scratch this phase will mostly be similar to the development phases of most Web Applications and even to other software products. If the project is based on a CMS tool this phase consists firstly on the installation of the CMS tool in the Web server and then the configuration of the desired modules (since CMST usually come with a set of modules already included, which perform specific functions) and then the development of new modules. In this phase the Graphical designers will work in parallel to define the layout of the system, which the developers then will apply, together with the necessary features. The testing sub phase does not necessarily comes after the development phase, depending on the methodology adopted for testing it occurs in parallel with the development. Nevertheless, after the development is finished some testing is necessary before the deployment of the system. The deployment can either represent the transference of the user credentials to the client or an actual deployment of the system in the client's Web server, in which case the development takes place in a testing server. The deployment is the last sub phase of the Setup and, at this point the, structure and the functionalities of the Content Management System should be ready, and the only thing missing should be the content. After the Setup phase the developer team passes on to the client the information of the administrator account, so that they can take over the system and manage it as they want.

- 2- **Content Production phase:** This second phase starts right after the deployment and as it can be seen in Figure 1 contains three main actions. In opposite to the Setup phase the Content Production phase does not have a determined finishing point, lasting during all the life of the system. This also does not have any sub phases in its constitution, but it has different blocks of actions associated with it. These actions may be more than those represented in the figure and may vary with the requirements of each system. Nevertheless, the ones that are represented are enough to characterize this phase. The system management actions are performed by the administrator, who is now someone from the company that owns the CMS. The actions consist in managing the user roles, the user permissions, the types of contents, and even the organisation of the system, in case that it has been implemented in the Setup phase. Depending on the requirements of the system the administrator might be able to manage some of the modules of the system. The content creation actions are normally performed by the Content Editor, but can also be performed by the Content Manager. The latter is also responsible for managing the content created by any of the users having permissions for that. The Content editor, in opposition, can only manage their own content and can even be limited to create content on some parts of the system. These permissions and limitations can be defined by both the content manager and the administrator of the system.

- 3- System Updates:** Even though in Figure 1 this phase shows up as being contained inside the Content Production phase, the System Updates is independent of it, and it is not a mandatory phase. In the drawing it is included to show that when it exists, it occurs in parallel with the Content Production phase. The System Updates are normally and ideally done by a development team assembled by the company that built the CMS, which can be the team that developed that specific CMS or a completely new team, and they take place when there are important things to be changed on the system that the administrator cannot change. This varies from platform updates, addition of new modules, deep layout changes, and other complex actions. As stated before, this phase normally occurs in parallel, normally in a copy of the actual system that then substitutes the system in the real Web server where the system is running. In this way the period of downtime is reduced and the changes can be tested without jeopardising the real system.

### 2.1.2 Content Management System Tools

CMS tools are a great alternative to systems built from scratch, since they make it easier and faster for companies to develop a new CMS. These tools normally come with a built-in structure for the back-end that allows an easy installation and configuration of whole system but also of the front-end. User profiles, user roles, section types, content types, and contact forms are some of the functionalities that normally are built into most of these tools and which can be turned on or off depending on the specifications of the project.

Currently, there are many different tools available, varying in technology used (PHP, ASP.NET, Perl, or other), in licensing type (Free to use, or paid), type of software design of the tool (Open or closed source), available features and other more specific characteristics. A simple search in an Internet search engine comes up with different websites that list many of these tools [20-26]. There are also different articles written by IT experts that try to advise and guide people through the choice of the best CMS tool for their needs, or if they really need a CMS at all [18, 27, 28]. Besides opinion articles, there are also some companies that produce reports where they analyse and compare different tools, like CMS Watch's[29] and Idealware's [30].

From the big number of existing tools might be interesting to look into a few tools as an example:

- Drupal [31]: Drupal is an Open Source project distributed under the GNU General Public License (GPL) [32]. This software written in PHP is free for download and comes with a Web installer making it easy to install a Content Management System in a Web server that supports PHP. This software package is maintained and developed by a large community of users and developers[33].
- Joomla [34]: Joomla is a free Open Source software thatlike Drupal is distributed under the GNU GPL license. Joomla has its origin in a fork that took place in 2005 of another well know CMST named Mambo [35]. Also like Drupal this software has its own community working on it. This community works in parallel

with the official Joomla Core Team [36], providing any user with different extensions and features that can be easily added to any installation of Joomla[37] .

- Umbraco [38]: Umbraco is an Open Source CMS based in the ASP.NET Framework, This CMS is considered to be one of the fastest growing of its type[39]. Umbraco is currently in its fourth version and is considered to be very flexible and easy to use.
- MOSS 2007[40]: Microsoft SharePoint Server 2007 is a very complete suite of server capabilities from Microsoft [41]that allows many different types of applications to be built with it. From intranet, to internet applications passy by CMS applications, all can be built with MOSS 2007. The applications built with MOSS are normally company focused and take advantage of the powerfull colaboration functionalites, that allow for documents produced with other programs from Microsoft's Office suite[42] to be shared between users of a MOSS application[43]. This tool is not an Open Source tool and is developed by Microsoft.

## 2.2 Web Hosting

A server is a computer designed to provide services to clients [44], and so, as the name suggests, a Web server is a server that provides Web services[45]. The World Wide Web Consortium (W3C)[46] defines Web service as “a software system designed to support interoperable machine-to-machine interaction over a network.(...) Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP (...)”[47]. Both the Simple Object Access Protocol (SOAP)[48] and the Hypertext Transfer Protocol (HTTP)[49, 50] are sets of rules which computers use to communicate with each other across a network[51]. For a CMS to be available all over the world through Internet access it must be deployed, installed or created in a Web server that provides a Web service capable of handling HTTP requests[52]. This way it is possible for client computers to connect to the Web server and, through an HTTP request, fetch the desired file.

When a website is loaded into a Web server, usually through File Transfer Protocol (FTP)[53, 54], it is said that it is hosted in that Web server. This is normally called Web hosting[55] and is one of many different Internet Hosting Services available, like E-mail hosting or DNS hosting[56].

Even though most companies have their own Web site, it is not very common for companies to have their own Web server. This is due to the cost of having and maintaining a Web server, which would make it mandatory for the company to have at least one dedicated machine, a fast internet connection to handle all the HTTP requests and a team of skilled people that is able to keep everything running smoothly. This is why most companies try to find another alternatives to host their Web sites, like hosting in the space provided by their Internet Service Provider(ISP) or by making a contract with other Web Hosting Providers[57].

There are many companies providing Web hosting services and there are also different types of Web hosting. The existing types of Web hosting are normally characterized in terms of



the disk space provided, the bandwidth available per month (which in fact is related with the quantity of data transferred available per month [58]), the database and other software support, and the required skills to make use of it [59]. The price of each Web hosting is also different from one type to another, increasing with the advantages and functionalities.

Any quick search in an internet search engine will lead to different Web sites that contain very similar lists of Web hosting types [59-64], from which the most interesting seem to be:

- **Free Web Hosting:** As the name states this type of web hosting is free. Most of the times free Web hosting providers provide reduced disk space, bandwidth, database support and even software support. Besides that, some of those providers also include advertisement in the users Web site;
- **Shared Web Hosting:** This type of hosting is said to be one of the most common types[60] and is characterized by the fact that the same server is shared by different users, including sharing the server's RAM memory and Disk Space. Shared web hosting is one of the cheapest solutions and it still provides good conditions[64]. Nevertheless, the prices of this type of hosting can also vary with the choice of characteristics. And since in this solution the user does not have the possibility of adding any software that he wants, he should choose the characteristics wisely and carefully. The shared characteristic of this server can bring some security risks, since there are many different Web sites on the same server[59];
- **Virtual Private Servers Hosting:** A Virtual Private Server is also referred to as **Virtual Dedicated Server** and consists on a real server partitioned in multiple virtual servers that, as far as the user sees, work as normal dedicated server. The difference of this type of hosting from the previously one is that, in this case, the resources like RAM and Disk Space are not shared with other users. But, since the virtual servers are running in the same server the latter's physical resources are actually shared, although when the server is partitioned in different virtual servers, those virtual servers are started with a specific amount of RAM and disk space that is used only by it. So, when a user rents a virtual private server, the look and feel are the same as if it had rented a dedicated server. The way these virtual servers are set allows the hosting client to have complete control over it, with reduced security risks, when compared with the previous hosting type. Even though there is a virtual separation between the resources, the physical resources are still shared and if something happens in some of the other servers it might compromise the other virtual servers. Sometimes this might happen to the greed of the service provider who believing that the clients won't use all the resources that they rent, might oversell virtual servers and risk every server's security. Another limitation of this type of hosting is related the inexistence of the possibility to expand the resources available over that capacity[65, 66];
- **Dedicated Web Hosting:** A dedicated hosting solution consists in renting the complete computer server instead of just a portion of it. This way the hosting client can take advantage of all the resources and capacity of the computer. This

makes it one of the most expensive solutions in terms of Web hosting. It is also a solution where the client normally has root/administrator access in the server and so he can configure it as he likes and he can install whatever software he desires. This solution makes it necessary for the company to have some skilled personnel to manage the server. Even so, most of the times there is still a level of action from the hosting provider in terms of hardware maintenance, including substitution of malfunctioning pieces. Sometimes these interventions from the hosting provider might be taxed separately and this should be taken in consideration when choosing a solution. This solution is normally chosen by larger companies that need more flexibility in terms of hardware and software resources, and that encompass in their human resources skilled personnel. For a web development company this is also a good solution because it allows to have many of its on-line products in the same server, in such a way that they can manage the server load depending on the popularity of each product, and their specific needs[64, 67];

- **Reseller Web Hosting:** The reseller solution allows the hosting client to become itself a hosting provider, partitioning and renting the web space that it has rented. This is not a regular hosting type and it is not an immediate alternative to the other types of hosting [68];
- **Collocation Web Hosting:** A collocation hosting service is somewhat similar to the dedicated hosting solution, but in this case, the computer server is not rented to the hosting provider but it is actually from the hosting client. The client's server stays in the hosting provider's premises, which normally are more appropriate than the client's own premises. In this situation the client is responsible for all types of maintenance of the server, from hardware to software, so it would be necessary for the company to have skilled personnel. This fact and the need to buy the hardware and any updates that become necessary are some of the reasons that makes this solution more expensive than most of the others [59, 69];
- **Cloud Hosting:** In the last few years a new solution for Web hosting has been born, and has been named Cloud Hosting [63]. Cloud hosting is a solution based on the services provided by Cloud Computing. Due to its novelty and complexity cloud computing will be presented and explained in the next section.

## 2.3 Cloud Computing

### 2.3.1 The Cloud

In the IT world there are different entities that take in their hands the mission of developing standards and defining this or that IT concept. But most of the times these entities are created to meet a necessity, and sometimes are not universally accepted. The absence of a regulatory entity

is one of the reasons why any new IT concept or term gets surrounded by confusion and discussion about its definition.

Another reason for the general confusion is the existence of many different actors in the IT theatre. With the expansion of the Internet and with the rise of Web 2.0 (another not entirely defined concept), the flow of information grew beyond expectations. As it was presented in the previous section, publishing content in the Internet was never so easy, and so it is no longer something only available to IT professionals. This made it even harder to establish an agreement on what is what, and how should it be defined. IT professionals, computer scientists, company representatives, and even anonymous folks now have a word about every and anything.

Besides this “freedom of speech”, also the different agenda of the different actors might influence what they have to say and what they find important for everyone else, making it a real challenge to establish a simple and perfect definition for every new buzzword or hype creating concept.

The Cloud is a good example of a buzzword to which there is not a widely accepted definition, as a list of twenty-one definitions from IT experts, that has been assembled by Jeremy Geelan in a blog post, shows[70].

When there is no unique definition, it is important to go through different definitions to try to understand what something is. This is why three different definitions or, at least, points of view about the cloud will now be explained.

- Vaquero *et al*[71], in an article entitled “A Break in the Clouds: Towards a Cloud Definition”, go through many of the existing definitions of the cloud in an attempt to build a consistent and complete definition of the cloud. In the article they refer which actors they think that participate in the cloud and what are the main services available. After their analysis they end up with the following definition: “Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs[Service Level Agreements]”. They also find it important to establish the differences between the cloud and the Grid, which is another IT concept that does not have a widely accepted definition.

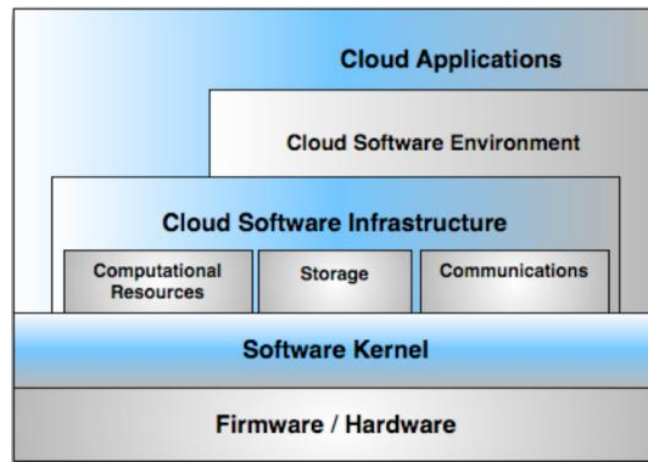


Figure 2 Youseff et al Cloud Ontology [71]

- The second document analysed is a presentation by Youseff *et al* [71] entitled “Toward a Unified Ontology of Cloud Computing”. In this document the authors present an ontology which they believe better represents the cloud and cloud computing, a computing style that leverages the characteristics of the cloud. Their proposed ontology defines the cloud as being divided in five layers, as can be seen in Figure 2 Youseff et al Cloud Ontology [71]. They defend that the five layers are organized in a stack like structure, where a higher position in the stack means that that layer of services can be composed of the services of the underlying layer. They also define that the cloud services belong to the same layer if their level of abstraction is equivalent, this being denoted by the target users of that layer. The defined layers are: Cloud Application; Cloud Software Environment; Cloud Software Infrastructure; Software Kernel; and the Firmware and Hardware represented by Hardware as a Service.
- The last definition analysed is from a presentation by Dan Stanzione [72], entitled “So, What is a Cloud?”. Since the two first points of view are quite similar from each other and this third one is a bit different, this analysis will be deeper to better understand the differences presented. Dan Stanzione starts by defending that some years ago there has been a fork in the computer systems community from which resulted two different groups, one formed by the so called “technical” computing community (representing science and engineering simulation) and another named “enterprise” computing community (representing business processes, offices and classrooms). Each of these groups focused in two different problems of computer systems, the first focused in solving the problem of processors being too slow to solve some of the problems and the latter trying to find ways for the servers to satisfy the necessary clients with reliability. Nevertheless, both groups attacked the problem through concurrency. In the author’s words “the S&E [science and engineering] computing community began down the road of supercomputing, and eventually settled on parallel computing as the answer, resulting in today’s high performance computing [HPC] systems” and “the enterprise computing community began down the path of failover and redundancy, resulting in today’s massively parallel utility computing systems, of which the ultimate evolution may be the cloud”. Nevertheless, the author thinks both groups ended up in the same place, but with small key technical differences and with the ability to solve very different

problems. After going over some details of the different paths taken, the author comes up with some characteristics that he believes to define the cloud, dividing them into characteristics he and his team are sure about, some that they think they are sure and even some that he knows other people with argue with him about. The first group of characteristics are that clouds are remote, scalable, and dynamic; the second group includes the fact that they are based in virtualization, provide an High Level API, and that they are loosely coupled; the last characteristics are the requirement of “a large scale, distributed storage system that abstracts how data is distributed and accessed”, and the one they believe to be more important that is related to how the cloud provides concurrent access to large data volumes. Summing up, Dan Stanzione believes that a cloud is not about computing, but yet “a cloud *\*is\** smart data”. This is probably the main point that settles the difference between this point of view and the previous two. But the author presents a cloud architecture to support his definition. For him, a cloud should be made of “Lots of systems; A storage mechanism that distributes data among the systems, with no large central storage; [and] An API that abstracts systems and moves work to data”. The author ends up the presentation by making the parallel between the clouds and HPC system. Whilst the clouds do for big data what HPC did for big simulation.

As one can see, there is not a consensual definition of what is the Cloud, even though all the definitions meet in some points. Nevertheless, it is not the objective of this project to define it. But it is important to assume at least one definition for what it is. As it was explained, this project is being developed in a Web development company, and it is focused in the analysis of different alternatives for CMS Projects. These systems are meant to be a communication channel between a company and its clients, and for small to medium sized companies they do not generate huge amounts of data. So, for a company like Tam Tam, the cloud presents itself as a possibly cheaper, easier to use and manage, hosting service. From this perspective, the differences in the definitions are not critical. Tam Tam is not seeking to develop its own cloud, instead it might use some of the services provided by the existing cloud services providers (which will be introduce in a further section). Having this in mind and considering that the first two definitions have the most points in common without any notable divergence, they will be regarded as good definitions of the Cloud. For what it matters, the differences (mostly noted by the detail of the second document on the lower layers) are not critical for the purpose of Tam Tam.

### **2.3.2 Cloud Computing**

In a blog post Martin MC Brown[73], an IT expert, points out what Cloud Computing (CC) is not, stating that it is not meant to solve just one type of computing problems but, instead, it is meant to be used to solve different computing problems or issues. CC can also be said to be the use and implementation of services provided by the cloud, leveraging the infrastructure to better serve their clients. These CC services use the same business model explained in Vaquero *et al*'s[74] definition of the cloud.

In their articles, both Vaquero *et al* and Youseff *et al*[71] identify different users/actors that interact with the cloud and with the cloud services. These are the Service Providers (SPs), who base their business in the provision of software services, the Infrastructure Providers (IPs), who provision the necessary computing infrastructure to host the services of the first, and also the normal users who use the services provided by the SP's.

Regarding the services of the cloud, their division is not exactly the same. Nevertheless, they seem to agree in three types, characterized by the virtualization of specific resources, in Vaquero *et al*, and by the layer where they belong to, in Youseff *et al*'s document. These types are:

- 1- Infrastructure as a Service (IaaS), consisting in computing resources managed by the IPs who, through virtualization, are able to create specific computing solutions to meet the resources demands (both storage and processing resources) of the clients, the SPs;
- 2- Platform as a Service (PaaS), which consists in the provision of the software platform where systems run on. In PaaS the sizing of the necessary hardware resources is said to be made in a transparent way;
- 3- Software as a Service (SaaS), consisting in software applications that instead of running locally, they run on the cloud using its resources. These services are normally available to a wide variety of users, and are developed by the SPs developers.

Youseff *et al*'s view places SaaS in the first layer, PaaS in the second layer, and IaaS in the third layer of the cloud infrastructure. These last authors in their presentation make a further separation in the case of IaaS in terms of the resources provided, dividing what Vaquero *et al* assume as IaaS in three different types of service: Data as a Service for storage, IaaS as memory resources, and Communications as a Service for the communication resources of the cloud. This division brings a level of detail that does not seem necessary in this context, so it will be assumed that IaaS covers these three services.

### **2.3.2.1 Advantages**

The different types of services provided by CC share the same general advantages of the cloud.

Cloud Computing makes it possible for companies, or even single clients, to rent theoretically infinite computing resources in such terms that they only need to pay for what they actually use, and with the possibility of scaling it up and down based on their demand, making it easy to understand why there is such an hype around cloud computing.

The flexibility provided by Cloud Computing is very interesting and as a concept seems to be a good choice for many of the companies' normal computing needs, either short or long term needs. The most obvious and maybe also the most important impact is a reduction in IT costs. Because, even if the prices of renting the necessary resources are high, they would hardly compare to the added costs of buying, installing, and maintaining a company's own hardware. In the case of a company already having its hardware, the flexibility of paying only for what is used and being able to scale up and down as necessary, makes cloud computing a favourite in terms of IT costs, because IT hardware needs maintenance and substitution from time to time. This is also true when comparing with one of the most common solutions used by smaller and

medium datacenters. In most datacenters the rental agreements are not as flexible as this solution and so, most of the times, there are resources that are not totally used. And when they are not enough scaling up is not exactly a fast process.

When going through the advantages of cloud computing, there is the idea that it is the perfect solution for all the IT needs of a company. But nothing is perfect, not even in IT, and so it is important to understand that these considerations are made in a general and abstract situation. It is of utmost importance that companies carefully study all the alternatives available to them. It is also important to keep in mind that this is just the good side of cloud computing. These advantages are the main reason why there is so much expectations about what will cloud computing bring to the future of IT and how will companies leverage it. The possibilities seem enormous, but it is still too soon to know. But besides the obvious advantages, there are also some concerns on the IT community. Everything comes at a price, and not only a financial price. If a company is thinking about moving their Web site to a Cloud there are some risks that it should take in consideration before deciding, to avoid complex problems.

### **2.3.2.2 Risks**

The risks associated with Cloud Computing are not necessarily impeditive but need to be accounted for and analysed very carefully before any move is made[75].

The base of all the risks behind Cloud Computing is associated with the move to a third-party infrastructure, be it to use computing power or storage capacity, or other type of service. This move brings with it some loss of control, lock in and privacy risks.

The loss of control over what is going on in the cloud has two different sides. One is the technical aspect of it. If some part of the cloud infrastructure fails there is not much that a company's IT personnel can do, besides waiting for the cloud computing provider to fix the issues and restore the system. The other side of it is related with Service Level Agreements (SLA). This deals made between the client and the cloud computing provider are still a bit unclear and vary from provider to provider and even within provider's different products, as is pointed out by Neil McAllister[76]. It is important for companies that use CC services that they do not overlook the SLA provided by the providers. This is even more critical if these companies are providing services to their own customers that require SLAs. In which case they will have to carefully plan their SLA to synchronize it with the cloud service provider's SLA[76, 77].

While explaining one of the sides of the loss of control, another problem associated with CC, the availability problem, was briefly approached. Even though most companies ensure an availability rate of over 99%, with some of them even promising 99,999%[78], the truth is that some problems of availability have been registered already by some vendors, like Google and Amazon[79-82].

Privacy concerns also arise with CC. Companies handling critical data, like banks and insurance companies, probably will not use cloud computing services. These companies usually have their own servers on-premises very well secured and cannot put their data at risk. But these are not the only companies having data that should be kept private. Many smaller companies have information about their products and clients in their Web applications or in their databases and they are not very fond of the idea of moving that data to other companies' servers. Besides

this trust problems there are also other privacy issues regarding the countries legislation over private data. If it is already hard for the countries to enforce their privacy laws in the Internet and in the datacenters inside their borders, it will probably be even harder to enforce any laws on data that can be in Europe now and in the United States just some minutes ago[83].

Cloud interoperability[84-87], or the lack of it, is another concern related with CC. Companies and developers are wondering if it will be possible to use some services of a vendor for an application and some other services of another vendor for another application, and still have both applications communicating with each other and exchanging information. Also the interchangeability might be an issue when moving to the cloud. It is not known if an application that runs in a providers' cloud will run in another providers' cloud. This possibility is widely expected so that a company does not get itself locked to a provider, having to accept any price changes that he makes, without being able to easily switch to a cheaper provider[88].

### **2.3.2.3 Future of Cloud Computing**

There are different opinions about the future of Cloud Computing, there are IT experts that think that it is being over hyped[89], while others think that it will not be a solution for the IT world, even though it allows for IT cost reductions[90]. And there are even some IT representatives who think that cloud computing is not mature enough for enterprises, as Craig Stedman from Computerworld[91] noted in an article[92].

Regarding the risks explained in the previous subsection, there are some who do not think that they are enough to stop companies from moving to the cloud. Mark Everett Hall defends that cloud computing can be safer than some on-premises solutions, stating that there are more known cases of problems in on-premises datacenters. But the absence of complete information about this makes it hard to say what is safer [93]. And Heather Havenstein, from Computerworld, analysed a study from Pew Internet & American Life that shows that the cloud is growing despite the privacy fears[94].

Something that has surprised and scared the IT community were the periods of outage that happened with Amazon Web Services in 2008[79, 80]. In May 2009, Google also suffered from some downtime caused by a mistake in their network configuration[82, 95, 96]. But, as with security, these problems can also occur in a normal datacenter, and there are not enough figures to say which the best solution is.

Despite the outages, the privacy fears and the maturity of the cloud, there are some people that foresee a long future for the cloud. Gartner Inc. the IT consultants have stated, back in 2008, that cloud computing would be as influential as e-business[97]. Irving Wladawsky Berger, in an article in AlwaysOn Web site, stated that the expectations for cloud services are quite high, expecting them to be simpler, safer and smarter than the on-premises services that are currently used[98].

In an effort to solve one of the concerns over the cloud, some cloud service providers, headed by IBM[99], gathered together and wrote the Open Cloud Manifesto. This manifesto aims to establish rules that the different cloud providers should follow. But as everything else around the cloud the manifesto did not have a consensus from all the vendors, with some of the bigger companies not signing it. Besides trying to define the cloud, this manifesto also tries to establish some principles for it. These principles include some proposal for providers to work



together in different fields to improve security, avoid lock in of users, among other things. An analysis made by William Hurley concluded that the manifesto might have been made in too much of a rush, leaving important details, like the definition of the concept of the cloud, out of it.

Regardless of having failed this attempt to define the cloud and define some principles to be followed by cloud providers, the Open Cloud Manifesto might have been a first step for some future agreement that will allow the IT industry to feel safer and more confident when moving to the cloud.

### **2.3.3 Cloud Services Providers**

Currently, there are already many different Cloud Services Providers, with different types of services. For this project the ones that matter the most are those providing hosting in the cloud, that can be seen as an alternative to data centers.

- **Amazon Web Services:** Amazon Web Services (AWS) is the cloud service provider of the well known bookseller Amazon. AWS was created in 2006 and provides a wide number of services, from computing to storage and e-business supporting services. In terms of hosting in the cloud, AWS has two main services, Amazon Elastic Cloud Computing (EC2), and Amazon Simple Storage System (S3). EC2 is a service that allows users to rent computing instances with pre-defined resources. In these instances they can run almost any operating system and software application. The configurations of the instances are named Amazon Machine Images (AMI) and there are already some pre-defined configurations from which to choose, nevertheless the users can define their own AMI according to their needs. After having an instance up and running, the users can upload and run their own applications or Web applications. The instances can be managed either with Elastic Fox, an extension for Mozilla Firefox, or with AWS Management Console. With these tools, users can easily scale up or down and initiate or shut down their instances. Amazon has launched in May 18th 2009 a new feature for EC2, called Amazon Cloud Watch. This service provides monitoring to EC2 cloud resources and allows the users to keep track of the resources usage and they can configure Amazon Auto-Scaling to scale their resources up and down based on the real-time needs[90, 91]. EC2 instances are virtual instances and, when they are shutdown, all the information stored in them is lost. To avoid the loss of data, Amazon provides another service that allows users to back-up the data from an EC2 instance into Amazon S3. Amazon Simple Storage Service, is a storage service that allows users to store and retrieve data in Amazon's cloud. AWS has two different geographical locations for its services from which users can choose, the United States of America and Europe[92].
- **Azure Services Platform:** Azure Services Platform is a cloud computing platform from the giant of computers, Microsoft. This platform is still under development, but developers can try it out already. Azure can be described as Microsoft's Cloud and it has a set of developer services that can be used by developers to develop

their own applications, either by using one or a combination of more of the available services. These applications can be Web applications, applications running on connected devices (like PCs or servers), or hybrid solutions. Currently, only applications developed in managed code languages supported by .NET are able to run in Microsoft's cloud. But in the official Web site it is stated that more languages will be supported. This platform provides an interoperable environment based in open standards, supporting multiple internet protocols, like HTTP, REST, SOAP, and XML. The base of Azure Services Platform is Windows Azure, which works like the operating system of the platform and, as such, it is the base of Azure Services Platform's structure. On top of it there are several different services that developers can use in their applications. Some of the available services are:

- services to handle user data and applications resources, named Live Services;
- services that extend Microsoft SQL Server into the cloud, the Microsoft SQL Services;
- services that include access control, service bus for communications across applications, and others, called Microsoft .NET Services.

For the moment Microsoft does not yet provide a relational database in this Platform, but it is working on it. Microsoft provides a toolkit to develop applications for its cloud that can be used with Microsoft Visual Studio. Currently, there is no information regarding the prices of running applications on Azure, but Microsoft stated that the prices would be very competitive, at least in relation to Amazon Web Services[93, 94].

- Google App Engine: Google App Engine[95] is a development and hosting platform from Google. This computing platform, which can be seen as Google's cloud, provides the users with an environment where they can publish and host their Web applications. Running a Web Application in Google App Engine is free, as long as the used resources stay under a determined amount. This amount consists of "up to 500 MB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month". If a user wants to expand its Web App resources he can enable billing, which is done by measuring the resources used. The users can also define a daily budget for their application. There are also some limits imposed in terms of Web applications per user account. Currently, this engine has a Java Runtime Environment and a Python Runtime Environment available, which allows Web Applications written in different programming languages to be run. This environment also has available something that is called Datastore, which is a data storage service featuring a query engine and transactions. This Datastore does not work like a relational database but Google ensures that is "strongly consistent and uses optimistic concurrency control". Google's cloud also provides some services that can be used by the users in their applications like URL Fetch and Image Manipulation, among others[96].

Google also provides a free Software Development Kit (SDK) to develop to Web applications to its app engine[97]. Google's cloud is still under development and new functionalities and services are expected to be released in the near future.

## **2.4 Summary**

This chapter started by presenting Content Management Systems the basic concepts behind them, and their overall characteristics. Special focus was placed in a CMS's lifecycle and in the user roles associated with the different stages of it. Some of the most popular tools to build CMSs were presented.

This chapter also introduced Web Hosting. Focusing on its importance, and its characteristics. The existing types of Web hosting were listed and explained. From these types one was of particular importance to this project, Cloud hosting. This type of hosting is the application of Cloud Computing to Web hosting.

The last section was dedicated to Cloud Computing. It started by presenting and analysing different definitions of Cloud, and by stating which definition was taken into consideration during the development of this project. In this section there it was also made an overview of Cloud Computing, its advantages, its risks, and the future perspectives of this style of computing. Finishing this section some Cloud services providers were introduced as well as some of their products.



## Chapter 3

# Multi-Criteria Decision Analysis

*In any moment of decision the best thing you can do is the right thing, the next best thing is the wrong thing, and the worst thing you can do is nothing.*  
Theodore Roosevelt[100]

This chapter starts by introducing Multi-Criteria Decision Analysis. An overview of this field of study is made, and some methodologies are introduced to prepare the reader for the following chapters.

### 3.1 Multi-Criteria Decision Analysis

MCDA is a field of study that was created in the decade of 1960[101] and is part of the wider field of Operations Research. At first, MCDA was named Multi-Criteria Decision Making (MCDM) but the word ‘making’ in its name was thought to be a possible cause of misinterpretation, since the aim of it was not to make decisions but instead to help others make decisions. In order to avoid misunderstandings, the word ‘making’ was substituted by the word ‘analysis’, and sometimes ‘aiding’. This change was mainly made among the European academics of MCDA, so there are some papers where the term MCDM is still used, causing some confusion at times[102, 103] In its 40 years of existence MCDA has developed to become a full-grown branch of Operations Research[104].

MCDA’s academics and researchers “are concerned with the design of mathematical and computational tools to support the subjective evaluation of a finite number of decision alternatives under a finite number of performance criteria, by a single decision maker or by a group.”[104]. In Kevin Pietersen’s words [105]: “The general objective of MCDA is to assist a decision maker or a group of decision makers to choose the best alternative from a range of alternatives in an environment of conflicting and competing criteria.”.

These definitions and concerns should be enough to clarify any doubts in relation to MCDA, nevertheless there are still some myths that sometimes are associated with it. Belton and Stewart[106] presented and dispelled these myths:

Myth 1: MCDA will give the 'right' answer;

Myth 2: MCDA will provide an 'objective' analysis which will relieve decision makers of the responsibility of making difficult judgements;

Myth 3: MCDA will take the pain out of decision making.

Although it would be very interesting if it was true, that is not the case. And if even for small day-to-day decisions there is not always a single right path to go, one can imagine how much rare that is in a more complex decision. Having to consider many criteria, many options, the opinion of many decision makers, and knowing how complex is the human mind, one can see how difficult it would be for a mathematical or computational tool to substitute the Decision Maker and reach for the perfect decision. But this cannot be seen as a flaw of MCDA, because MCDA never aimed at being the compass to the perfect decision. As the above definition stated MCDA and its researchers aims at aiding decision makers in their function of making decisions.

Since provide decision aiding is one of the main goals of MCDA, it is important to understand what this is. Bernard Roy *et al*[107] define decision aiding as “the activity of the person who, through the use of explicit but not necessarily completely formalized models, helps obtain elements of responses to the questions posed by a stakeholder of a decision process”.

José Figueira *et al*[101] identify the contributes of decision aiding: “analyzing the decision making context by identifying the actors, the various possibilities of action, their consequences, the stakes; organizing and/or structuring how the decision making process unfolds in order to increase coherence among, on the one hand, the values underlying the objectives and goals, and, on the other hand, the final decision arrived at; getting actors to cooperate by proposing keys to a better mutual understanding and a framework favourable to debate; elaborating recommendations using results taken from models and computational procedures conceived of within the framework of a working hypothesis; participating in the final decision legitimization.”.

These contributes are one of the reasons why MCDA has so much potential and why it is used by many companies to solve different problems. These problems can be very different and can fall into different categories. Valerie *et al* [106] identified some categories based in a set of characteristics that they had identified. These categories are: “One-off vs. Repeated problems; Number of stakeholders; Status and influence of the client; The ‘problematique’; Range of available alternatives; and Facilitated vs. Do It Yourself.”. Each of these categories influences the decision making in different ways. These influences must be accounted for during the process of multi-criteria decision making, to better help the stakeholders, also called the decision makers (DM), to understand the different options and to develop coherent preferences over these options.

In a report named “DTLR manual on Multi Criteria Analysis”, from the former United Kingdom Department for Transport, Local Government and the Regions (DTLR)[108] the process of application of MCDA divided in the following stages:

- 1- Establish the decision context
- 2- Identify the options to be appraised;

- 3- Identify objectives and criteria;
- 4- 'Scoring'. Assess the expected performance of each option against the criteria. Then assess the value associated with the consequences of each option for each criterion;
- 5- 'Weighting'. Assign weights for each of the criterion to reflect their relative importance to the decision;
- 6- Combine the weights and scores for each option to derive an overall value;
- 7- Examine the results;
- 8- Sensitivity analysis;

If the sensitivity analysis shows that the results of the other stages are vulnerable to the analysis, the previous stages should be repeated, especially from stage four until a new sensitivity analysis stage. It might also be important to look into different options not yet considered. As one can see, these stages do not incorporate the act of decision making, because as it has been stated, this is not the objective of MCDA.

In the 40 years of life of MCDA, different schools of thought have been born and have evolved in parallel with each other, promoting discussions around the theme of MCDA and different approaches have been taken by each school [102]. Although some years have passed since the formation of most of these schools there has been some confusion about what are the different schools. Different authors have identified different schools.

Belton and Elder[109] identified three schools of thoughts: one referred to as MCDM/Multi Objective Linear Programming (MOLP) which origins would situate in extensions of linear programming to problems with multiple objectives; another school from the "US dominated field of decision theory, incorporating multi attribute value/utility theory"; and the French led school. Herath and Prato[110] made a differentiation based in the focus of the approach, either based in the value and/or utility function or an approach based in the quantitative risk. In this case they do not make a geographical separation.

This lack of consensus is referred by Freerk Lootsma[104], in his book named "Multi-Criteria Decision Analysis via Ratio and Difference Judgement". He presents his view on the different approaches followed by MCDA research in "order to support the decision makers in their attempts to identify a preferred alternative, to classify the alternatives, and/or rank-order them". He defines four different approaches: the descriptive approach, the normative approach, the perspective or clinical approach, and the constructive approach. Lootsma then points out some articles that associate the constructive approach as the French school, and the normative approach as the US school. He also refers an article where Roy and Vanderpooten [111] present and analyse the European school and also state that the French school and the European school are the same school. In the referred article of Roy and Vanderpooten, published in the Journal of Multi-Criteria Decision Analysis, there is a comment from Lootsma where he criticises some of the considerations made on the article. Lootsma points out that there does not seem to exist enough information to state that each schools is completely independent of others, and he wonders if it is possible to say that European researchers are not influenced by what is developed in the US and vice versa. He also questions if the different methods, usually associated with specific schools of thought, have so well defined borders and if they are developed to help only the decision makers of the geographic region associated with the school

where they were developed. Lootsma also puts some attention in the subjects of MCDA research, stating that maybe research should also focus in the different types of management and not only on the differences between the methods or between the approaches. With this, Lootsma does not mean that there should not be different branches in MCDA, but instead that they should not focus on cultural differences, since these are not mirrored in the resulting methods. Lootsma also points out the need to find a “shared view on how human preference and human value judgement should be modelled” and to define or agree on “how decision processes in groups proceed via negotiation and power games”.

Even though Lootsma’ ideas, some recent books still use the notions of the American and French (or European) schools[101].

Nevertheless, it seems natural that these differences exist knowing that, in terms of science and research, 40 years of existence is still the youth time for most of the fields. More important than these disputes is, probably, to notice that there are numerous articles reporting the use of MCDA methodologies in different environments, from public to private, showing that the main objective of MCDA, to aid decision makers, is being achieved in different ways.

## **3.2 Methodologies**

The most well known MCDA methodologies can be grouped into three different categories like Freerk Lootsma suggests[112]. The first category includes the methods closely related with Multi-Attribute Utility Theory (MAUT), the second category includes the methods that derive from the Analytic Hierarchy Process (AHP), and finally, the third category encompasses the methodologies that revolve around ELECTRE, an outranking methodology from the sometimes called French School. These methodologies are also simply known as outranking methodologies[112]. Vassil Vassilev *et al*[113] places AHP methodologies in the same group of the Multi-Attribute Utility Theory, and points out a third, or fourth in this context, a group composed of interactive algorithms, which, the authors state, are focused in solving “multi-criteria decision analysis problems with large number of alternatives and a small number of criteria”.

There are a few different methodologies in each one of these groups. In this report only a few will presented.

### **3.2.1 Multi-Attribute Utility Theory**

In a problem with different criteria and different alternatives the MAUT methodologies try to define which of the different alternatives scores the best in the different criteria. In general terms this is done by defining a hierarchy of the different criteria and assigning different weights to each criterion. After this has been done it is measured how well each alternative scores with each criterion. This is calculated through an utility function that can be linear or non-linear. It is the job of the analyst who is applying the methodology to find the utility function. These methodologies follow a set of six axioms “which are meant to eliminate inconsistencies and suboptimal choices when it comes to trade-offs and uncertainty.”[114]. These axioms are the following:



- Completeness of complete order:  $X$  is preferred/less preferred/equally preferred to  $Y$ ;
- Transitivity: if  $X > Y$  and  $Y > Z$ , then  $X > Z$ ;
- Probabilities exist and can be quantified;
- Monotonicity of Probability: decision maker prefers a larger probability of a good outcome than a smaller probability of a good outcome;
- Substitution Independence: preferences are linear with respect to probability.

There are many methodologies that fall in the MAUT methodologies group[113]. In this report only one methodology of this group will be introduced, since it seems enough to understand with a bit more of detail the general concept of these methodologies.

## **SMART**

The Simple Multi-Attribute Rating Technique (SMART) is a simple MAUT methodology. In this methodology the ranking of the criteria, and then of the alternatives in relation to each alternative is done directly. The direct ranking can be done in many different ways. One of these ways is to start by identifying the most important element of the set currently being ranked and assigning it a value of 100, in a scale from 0 to 100. After that, all the other elements are assigned a value in the same scale considering their relevance in relation to the most important element [115]. After having the ranking the analyst uses the utility function that he previously defined to calculate the utility of each alternative.

This methodology was developed in the late 1960's and early 1970's by von Winterfeldt and Edwards [116].

### **3.2.2 AHP**

The Analytic Hierarchy Process (AHP) is a MCDA methodology developed by Saaty [117]. This methodology bases the decision aiding process in building a hierarchy to model the problem to which AHP will be applied. This hierarchy contains the goal or the mission statement, the criteria, and the alternatives of choice. Before the hierarchy is constructed it is necessary to verify if the different criteria are independent of each other. The use of dependent criteria in the decision process cause any alteration to these criteria to have a double impact in the prioritization of the criteria and also of the alternatives. The hierarchy is evaluated through the derivation of ratio scale priorities from pair wise comparisons, or judgements. After the hierarchy is defined and the information regarding the alternatives is known it should be verified if any dominated alternatives exist. In AHP an alternative A dominates an alternative B if A is better than B in at least one criteria, without being worst than B in any other criteria. The judgement assessment is made by questioning the Decision Maker about his preferences between each pair of criteria and afterwards between each pair of alternatives. This process for data input is very straightforward and is one of the most praised characteristics of AHP.

To assess the judgements from the Decision Makers, AHP uses the following scale (Table 1) defined by Saaty [117].

*Table 1 AHP Judgement Scale*

| <i>How important is A relative to B?</i>  | <i>Preference index assigned</i> |
|---|----------------------------------|
| Equally important   | 1                                |
| Moderately more important   | 3                                |
| Strongly more important   | 5                                |
| Very strongly more important  | 7                                |
| Overwhelmingly more important   | 9                                |
| Intermediate values used to represent shades of judgement between the five basics assessments | 2, 4, 6, 8                       |

The opposite preference (from B relative to A, for instance) is the inverse of the preference from A relative to B.

There is a wide number of methods that derive from AHP, trying to take advantage of its strengths, and avoiding its weaknesses. One of AHP's main weakness is rank reversal. Rank reversal is the "possibility that, simply by adding other option to the list of options being evaluated, the ranking of two other options, although not related in any way to the new one, can be reversed" [108].

## **REMBRANDT**

REMBRANDT (Ratio Estimation in Magnitudes or deci-Bells to Rate Alternatives which are Non-dominant) is a system for MCDA that was developed in the early nineties by Leo Rog [112]. This methodology is an alternative to AHP. This method uses a direct rating system which is on a logarithmic scale, replacing the scale used with AHP. Another difference from this method to the AHP method is in the replacement of the eigenvector-based synthesis approach for one based on the geometric mean to calculate the priority values from the comparison matrices[108].

### **3.2.3 Outranking methods**

This group of methodologies is said to have been founded by Bernard Roy during the late 1960s, when he developed the ELECTRE family of methods [118]. Following the methods from this family other have been created using the same concepts. All outranking methodologies share two major stages during their application. In the first stage an outranking relation is defined, and this relation is then used in the second step to make the evaluation of the alternatives that were considered [118]. During the process these methods also aim at eliminating the alternatives that are dominated. Here the notion of dominance is not as straightforward as the notion used in AHP family of methods, since in outranking methods weights are used to change the influence of some criteria over the others [108].

To compare different alternatives, four binary relations are used: "the indifference I (reflexive and symmetric), the weak preference Q (irreflexive and anti-symmetric), the strict preference P (irreflexive and anti-symmetric), and the incomparability R ( irreflexive and symmetric). The outranking relation covers these four relations." [113]. These outranking

relations are used to model and represent the Decision Makers preferences, and are obtained through their direct interrogation. Michael Doumpos and Constantin Zopounidis [118] point out that this way of modelling the preferences differs in two major issues from the one used in the MAUT methodologies. These issues are the non-transitiveness of the outranking relation and the fact that it is not complete.

A closer look will be given to the ELECTRE family of methods, which are the most recognizable set of outranking methods.

## **ELECTRE**

ELECTRE, which stands for ELimination Et Choix Traduisant la REalité, is a family of methods originated in the mid-1960's, and due to Bernard Roy [119]. These methods are concerned with identifying dominance, as explained in the above sub-section. These methods "seek to locate a subset of options, E, such that any option not in E is outranked by at least one member of E." [108]. The application of these methods can be divided in two different phases. In the first phase, concordance and discordance are defined for every ordered pair of options. Then, two thresholds are defined, one for concordance and another one for discordance. An option is said to outrank another option if some criteria respecting these thresholds are met. After this phase there should be a set with the promising options. Since in these methods the definition of the thresholds are in the hands of the Decision Makers, these methods are considered to leave a big part of the final decision to the Decision Makers [108]. The family of ELECTRE methods already counts with five evolutions of the original method. The choice of which ELECTRE method to use can be guided by a table defined by Bernard Roy [119].

### **3.3 Summary**

This chapter focused on MCDA, making an overview of this Operational Research branch. Also some methodologies developed by MCDA researchers were introduced.



## Chapter 4

# Problem Overview and Approach

*All progress is precarious, and the solution of one problem brings us face to face with another problem.*  
Martin Luther King Jr. [120]

This chapter is dedicated to the definition of the problems that this project aimed at solving. It is then described the approach that was taken to solve those problems.

### 4.1 The Main Problem

The importance of Internet in today's business world is undeniable. As of December 2008, the number of Internet users in The Netherlands was estimated to be 82% of its population[121]. This means that for a company in the Netherlands, no matter its size, one of the best ways to reach its clients is to be on-line (i.e.: to have at least a presentation Web page available through the Internet). Tam Tam's main clients are small to medium size companies, which requirements vary from a simple presentation Web page, to a complete intra and internet solution. To face this variety of requirements Tam Tam has divided their Human Resources in different Advisory Groups that would specialize in different types of solutions, without being totally independent from the other groups.

One of these Advisory Groups is the Content Management Systems group. This group is mainly focused in the development of CMS solutions. These solutions are normally developed with one of two CMS engines, or tools, Microsoft Office SharePoint Server (MOSS) and SmartSite. The developed solutions are normally hosted either on the client's own premises or in TerreMark's servers. TerreMark is a hosting company which owns a datacenter in Amsterdam. Tam Tam has been working with TerreMark for some years now and has most of its servers hosted at TerreMark's.

CMS projects are normally built for smaller companies whose requirements are simply to have a presentation Web site where they can promote the company and its products with the

ability to easily update the contents by employees without any knowledge of programming languages. These smaller companies are also the ones that normally suffer the most with global crisis like the one that has started in the end of 2008. The financial difficulties due to the crisis might make these companies cut on their budgets. And in the case of companies not focused in new technologies, or those that do not value Internet presence too much, might cut their IT budget. These potential cuts might mean that they do not identify the investment in a Web application as a priority, or that they will try to find the cheapest solution, sometimes overlooking the quality.

The scenario presented in the above paragraph is one of the reasons why Tam Tam has defined the problem in which this project is focused, in order to try to reduce the prices of its solutions while maintaining their quality. Another reason is Tam Tam's commitment to innovation.

The main problem is to help Tam Tam find an alternative solution for the development of their CMS projects. The resolution of this problem was divided in two sequential phases. The first phase is composed by research and definition of the different scenarios, and the second phase is the characterization and comparison of those scenarios, so as to help Tam Tam to find one or more alternative scenarios to its current approach.

To better understand the problem and its resolution, it is important to take a look at some constraints and details of both of these phases. The details about the process of resolution of the problem at hands are covered in Chapter 5, except for the comparison stage which will be presented in Chapter 7.

#### **4.1.1 Research and Definition**

The first stage of the resolution of this project's main problem was to search and analyse different technologies and tools that could be used to define a set of scenarios, alternatives to what is currently used at Tam Tam.

Each scenario has to be composed of a CMS tool and a hosting solution. Each of these components has to obey to some constraints.

Concerning to the CMS tools:

- Development at Tam Tam is mainly done with the .NET Framework, so only CMS tools built with this framework could be part of a scenario;
- Tam Tam already uses two proprietary technologies, MOSS and SmartSite so, besides these two tools, only Open Source tools should be considered.

In terms of hosting:

- Since Tam Tam already has most of its servers in a datacenter from TerreMark, they think that there is no need to look into alternatives of this type. They think that it would be more interesting to focus in cloud computing services.

### **4.1.2 Characterization and Comparison**

After defining the constraints imposed to the possible scenarios, it is time to characterize and compare them. In the characterization of the different scenarios there are some questions that have to be answered. These questions reveal some concerns and some things that Tam Tam's responsible would like to see covered by the characterization:

- Who will be responsible for the maintenance of the server where the applications will run?
- How complex will it be to publish an application, to access it, and to update it?
- Can different CMS applications run in the same Web server, sharing the same code-base or will it be necessary to make a new installation for each new application?
- How easy is it to update an application?

Some of the criteria used to characterize the different scenarios were inferred from these questions. It was necessary to define other criteria that represented as much information as possible about the each scenario, to facilitate the comparison and the evaluation of each alternative.

## **4.2 The Second Problem**

The second phase of this project is directly related with the comparison of scenarios identified in the first phase. The main problem was to find a methodology that could aid Tam Tam on the comparison of the different scenarios, giving them information and support to make a decision.

To solve this problem, different Multi-Criteria Decision Analysis methodologies were analysed. After choosing an adequate MCDA methodology, a software application was developed to implement the chosen methodology. This application was then used to make the comparison of the different scenarios, simplifying this process. Details about the process of choice of the methodology, as well as the implementation of the application, can be found in Chapter 6 of this report.

## **4.3 Summary**

This chapter started by explaining the context of the problem which this project tried to solve: helping Tam Tam define and compare different alternatives to the way their CMS Projects are developed. Afterwards it is presented the approach that will be taken to solve the problem. This approach is divided in different phases that are explained in detail.

A second problem that this project faced is also introduced in this chapter. This second problem was to find the best MCDA methodology to help Tam Tam with solving the first problem. It was also defined that an application would be built to implement the chosen methodology. The resolution of both of these problems will be described in the next three chapters.





## Chapter 5

# The Alternative Scenarios

*The absence of alternatives clears the mind marvellously*  
Henry Kissinger [122]

This chapter describes the steps taken to build the set of alternative scenarios for Tam Tam's CMS projects. It starts off with the details of the definition of the scenarios. Then it focus in the definition of the criteria used for characterizing the scenarios, and after that it proceeds with the characterization.

### 5.1 Scenarios Definition

The first step towards the solution was to define the different scenarios that could work as alternatives for the development of Tam Tam's CMS projects.

In this step there was a research phase, focused in understanding what possibilities Cloud Computing had to offer, such as which are the available services and the main providers. This research also aimed at finding different CMS tools that might serve Tam Tam's objectives. The research was made with the constraints described in Chapter 4 in mind and were defined in the beginning of the project. After the initial period of research current scenarios used by Tam Tam were looked into and some alternative scenarios, considered internally as possibly good alternatives, were drawn. At this point it was possible to make a first draft of the scenarios which also helped in focusing the research, and led to an initial analysis of some of the possibilities presented.

The described process will now be presented in more detail, dividing it into three parts.

### 5.1.1 Research

The research made in this step focused in possible Cloud services and in finding different CMS tools. These tools had to obey to some constraints, but the Cloud services' research was even more restricted, as it was presented in Chapter 4. These constraints and restrictions were defined right in the start of the project.

In terms of the Cloud services two providers were considered as the two alternatives to be explored for the hosting part of each scenario. These providers were Microsoft's Azure Services Platform and Amazon Web Services.

As the Cloud services providers, that were to be used in the definition of the scenarios, were already defined, the research on this subject was focused on the analysis of the specific Cloud hosting services of each provider and their characteristics.

In terms of CMS tools it was compiled a list of different tools:

- **Cuyahoga [123]:** an Open Source CMS tool built with the .NET framework;
- **DotNetNuke [124]:** an Open Source CMS tool built with the .NET framework;
- **Umbraco [38]:** an Open Source CMS tool built with the .NET framework;
- **WordPress [125]:** an Open Source publishing platform that can be used to build CMS projects; it is built with PHP,

All the tools except WordPress met the constraints presented in Chapter 4, namely being built in the .NET framework and being Open Source. Although WordPress does not meet the first of these constraints it was also listed because some interest for this tool had been shown. This interest was mainly due to its versatility and to the fact that it allows different applications being built on top of the same code-base. This characteristic is quite valued by Tam Tam because it saves a great amount of time whenever it is necessary to update the base of all the applications.

### 5.1.2 First Draft

In this part of the project the two scenarios that normally are used to build a CMS project at Tam Tam were analysed.

The scenarios presently used at Tam Tam are the following<sup>2</sup>, in no particular order:

- 1- In the first scenario the CMS application is built with Microsoft Office SharePoint Server (MOSS). The applications are installed in a Web server hosted a TerreMark running Microsoft Windows Server 2005, Microsoft Office SharePoint Server 2007 and with a database server running Microsoft Windows Server 2005 and Microsoft SQL Server 2003;

---

<sup>2</sup> Some projects might not follow exactly one of these scenarios because some companies have their own server structure on premises or some contract with a specific datacenter and so they prefer to host the application themselves. Nevertheless that is not too important for the problem at hands, because those are the exception.

- 2- In the second scenario, the CMS application is built with SmartSite. The applications are installed in a Web server hosted at TerreMark running Microsoft Windows Server 2005, and a database server running Microsoft Windows Server 2005 and Microsoft SQL Server 2003.

Based on the listed scenarios a draft of a possible set of scenarios that could satisfy Tam Tam's needs and requirements was built.

- 1- MOSS as the CMS tool and a Cloud hosting solution;
- 2- SmartSite as the CMS tool and a Cloud hosting solution;
- 3- Umbraco and TerreMarks hosting solution;
- 4- WordPress as the CMS tool in WordPress.com servers building different applications on top of the same account.

From the information gathered it was clear that Tam Tam was not in any way unpleased with the current tools that it uses. But having the objective of cutting in costs these CMST were paired with a Cloud hosting solution. From the list of CMS tools compiled and shown above, Umbraco was the only one that was included in this draft. This choice was made with base on the feedback from the team involved in a project at Tam Tam that used this CMST. In this draft Umbraco was put together with a TerreMark's hosting solution. In this case the cost reduction would be in the side of the CMS tool, since Umbraco is an Open Source tool. In this draft it was also included a scenario with WordPress. Although Wordpress based in the .NET f. This inclusion is due to the possibility of hosting applications in WordPress.com's servers and of developing different projects on top of the same account.

### 5.1.3 The Scenarios

The step that followed the definition of the above draft focused in getting some more information about different aspects of CMS Projects and how important can some of these aspects be. One of the aspects that was paid more attention to was the codebase characteristic of the CMST. A CMST that works with a single codebase allows different applications to be built on top of the same installation, running independently. One of the main advantages is related with the updates done to the codebase, that by being shared would save time. But this could also pose a danger, because if the update goes wrong all the applications running on top of the same codebase could suffer. So at that point it seemed that having a singled codebase would be appreciated but was not mandatory.

Another point analysed was the hosting solutions at TerreMark. Although their service is not perfect and their prices were a bit high, it seemed important to consider at least one scenario with it, so that it could be directly compared with the Cloud hosting solutions.

During this phase it was reached the following set of scenarios:

- 1- **Scenario 1:** MOSS as the CMS tool using the Cloud hosting services provided by Amazon Web Services;
- 2- **Scenario 2:** Custom CMS tool that would be built by Tam Tam using the Cloud hosting services provided by Microsoft's Azure Services Platform;

### 3- **Scenario 3:** Umbraco as the CMS tool using the hosting services provided by TerreMark.

The set of defined scenarios shows that the option of using WordPress was ruled out. The reasons behind this are related with the fact that it is not a .NET based tool, but also with the fact that in the solution presented in the draft, which seemed to be the best use for WordPress in this context<sup>3</sup>, Tam Tam would not have much control over the server's configurations and management. And even though it is desirable that these tasks are easy to perform, one of the reasons why Cloud hosting seems such a good alternative, is because it is not at all desirable to lose all the control from what is going on in the server, especially when there will be hosted different CMS applications from different clients.

In this final set the first scenario of the list is composed by MOSS as the tool for the development of the application hosted at Amazon Web Services. The choice of MOSS is easily explained. Since MOSS is one of the main tools used by Tam Tam in the development of different types of projects, including CMS projects, from what one can imagine that the level of skill of Tam Tam's developers with MOSS is quite high. Besides the familiarity with the tool the tool itself is very powerful and can be used to develop different types of projects, as it has been stated before, which can make it easier to upgrade or evolve a CMS application to a more complex type of application or even to simplify the process of integrating the CMS application with other companies, if needed. The choice of Amazon Web Services for the hosting part of this scenario is related with the different types of Cloud services provided by Amazon, the fact that is a provider already with some years of experience (it was launched in 2006), and also because this way a second Cloud Computing alternative can be analysed with more detail.

A custom CMS tool that would be built by Tam Tam's developers to run on Microsoft's Cloud platform composes the second scenario defined. The decision to analyse a scenario with a custom CMS is related with the thought that a tool built by Tam Tam itself would make the development of every new CMS project easier than with the current tools, with which the initial phase of development consists of removing and cleaning up the functionalities that come in excess and which normally are not necessary. The hosting solution for this scenario was chosen with the objective of looking with some detail into Microsoft's Cloud platform. One of the things that needed to be looked into was if an application needed to be built with any specific requirements, tools or services to run on Azure had, so that it would make it hard or even impossible to move it to other type of hosting if needed be. This was analysed in the characterization of the scenarios that will be presented in a later section. A scenario composed by a custom tool might seem to be positioning itself as the best and perfect choice since it can be built to meet all the requirements for a perfect CMS project, if that exists. Nevertheless the cost of developing a solution like this, the complexity of the development, and the time it might take can be strong arguments against it.

Umbraco, an Open Source CMS tool and the hosting solution currently used compose the last scenario defined. This tool had already been chosen in the draft presented in the previous sub section. It has been used by some of Tam Tam's developers and it has caught their attention, due to its simplicity. This tool was conjugated with TerreMark's hosting services because the

---

<sup>3</sup> Another possible use for WordPress is downloading the WordPress publishing tool from <http://www.wordpress.org> and installing it on a Web server. Instead of using WordPress.com's hosting services.

tool itself might cause sufficient cost reduction to make it unnecessary to move from TerreMark's servers to a Cloud solution. This way this type of hosting can be somewhat directly compared with the Cloud hosting.

From the draft to the final list of scenarios one can see that SmartSite was left out, this is in no way related to the quality of this tool, Tam Tam felt like it was not worth it to analyse a different scenario composed with it. And so it was decided to leave it for a future analysis, in case no one of the defined scenarios shows to be a better solution when compared with the scenarios that are being currently used.

## 5.2 Scenarios Characterization

The characterization of the different scenarios was based on a set of criteria that covered the most important characteristics that Tam Tam would like to compare. Before going into the scenarios details it is important to know the set of criteria and why each criterion was chosen.

### 5.2.1 The Criteria

The set of criteria used to characterize the scenarios was compiled through the analysis of some of the main characteristics of both CMST and Web hosting (including Cloud hosting services). Some of the criteria regarding CMST are based in the ones used in the CMS Matrix Web site to classify different CMST [28].

The criteria that were defined are the following:

- **Hours of development:** Average number of hours needed to build and deploy a CMS;
- **Codebase type:** In this context codebase type can be either multiple or single. If it is possible to build different and independent Web Applications on top of the same installation of the CMST, then the codebase is considered to be single, otherwise it is multiple;
- **Security:** This criterion evaluates if the CMST provides security functionalities like secure authentication, definition of privileges for the different users, among others;
- **Ease of use:** This criterion represents some characteristics of the CMST that make the resulting systems easy to use, like the production of Friendly URL's, the existence of a Spell Checker functionality, a What You See Is What You Get (WYSIWYG) text editor, among others. These characteristics or functionalities simplifies the addition of content to the system. So it is important if a tool already comes with some of these included;
- **Built-in Applications:** This criterion measures how complete is a CMST in terms of built-in applications. Some tools come with applications like a blog engine, or a forum engine which makes it easier to develop products that require this type of

applications. Nevertheless too much built-in applications can have a negative effect in the time that it takes to configure the tool to build a new CMS;

- **Added Value:** This criterion represents how well can Tam Tam leverage one of the scenarios in terms of clients' satisfaction in relation to the product built. It is a somewhat subjective criterion but it also gives some information regarding the scenarios;
- **Deployment:** Complexity of deploying a new Web Application built with the CMST;
- **Update Web Applications:** Complexity of updating a new Web Application;
- **Portability:** When building applications to run in a specific hosting environments there might be some constraints that make it harder to move it from that environment to an alternative hosting environment, requiring some changes to the application. This criterion represents those difficulties if they exist;
- **Scalability:** In terms of hosting there are normally an amount of resources available per application, sometimes it is necessary to scale those resources up or even down. This criterion represents the time it takes to scale the resources with a specific type of hosting;
- **Redundancy:** This criterion is also connected with the hosting solution, and refers to the existence of a server structure that ensures that if one server fails to provide it services there is an alternative server to substitute it. Redundancy of a scenario can be classified inexistent, partially existent or existent;
- **Load Balance:** Criterion to characterize the existence or not of tools that allow for the load balancing of the hosting solution. Load balance is important to handle situations where the number of requests made to a Web application is high enough to compromise the handling of those requests. Load balance allows for the management of the requests either by dividing them through different servers, if there are more servers available, or by putting some on hold while the server is busy processing other requests. Load balance of a scenario can be classified as inexistent, partial or total;
- **Backup of Application and Data:** Criteria to characterize the server in terms of means and solutions to backup the application and its data;
- **Backups Redundancy:** Besides the backing up of the data it is of some importance to have backup redundancy in case of something wrong happens with the first backup. This criterion can be classified as inexistent, existent and existent as an extra service;
- **Server Updates:** This criterion refers to the existence of periods of downtime associated with server software updates;
- **Server Maintenance:** The server maintenance is an important aspect of the Web hosting, this criterion represents the complexity behind the maintenance of the

server;

- **Launch cost:** The use of some of the software present in the scenarios requires the payment of licenses to the company that produces it. Launch cost represents these licenses as well as any other costs associated with the launching of a project in each scenario;
- **Cost per Project:** Not all the costs fall into the Launch Cost criterion, because some costs of building and running a Web application are charged monthly or yearly and so they should be divided by the average number of projects that is run in the same hosting solution. Tam Tam normally hosts an average of 45 Web Applications in the same server;

The defined criteria seemed to be quite complete in covering the different aspects of a scenario that would allow for a complete and thorough comparison of the set of scenarios. Next will be the actual characterization of the scenarios. This was done by filling the information of each scenario in relation to this set of criteria.

## 5.2.2 Characterization

Having the criteria defined it was time to proceed to the characterization of the scenarios. First, there will be some details about the server structure defined for each scenario and then the different criteria will be filled accordingly.

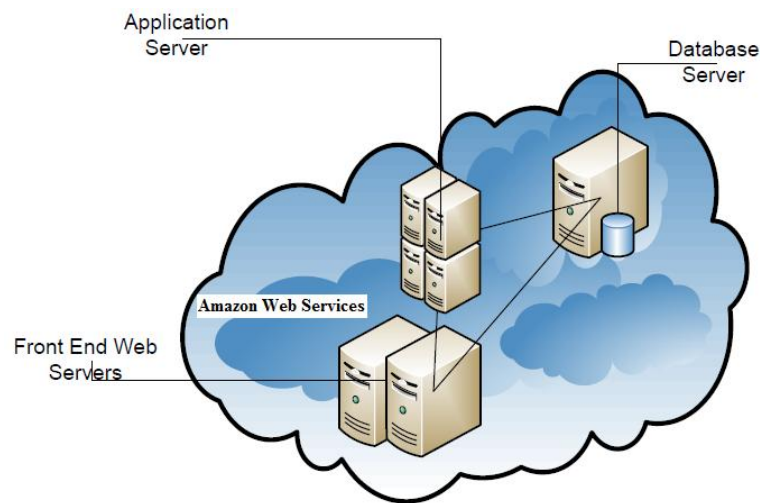
### 5.2.2.1 Scenario 1: Microsoft Office SharePoint Server 2007 + Amazon Web Services

The first scenario is composed by Microsoft Office SharePoint Server 2007 (MOSS 2007) as the CMST with the different applications being hosted in Amazon Web Services.

The server structure was based on the characteristics of the tool. There are different structures in which MOSS 2007 can be run, depending on the complexity of the applications. In the case of CMSs a medium server farm is normally considered to be sufficient, and it is the solution that Tam Tam normally uses for CMS projects. In average a medium server farm hosts 45 different applications. A medium server farm consists of four servers: two Front end Web server; one Application server, and one Database server (See Figure 3 Amazon Web Services – Medium Server Farm[119]. In a farm like this the Front end Web servers will be responsible for handling the HTTP requests, the Application server will host different SharePoint services like the indexing and search services, and the Database server will run the database that will support the applications. Both the Front end Web servers and the application server need to have MOSS 2007 installed on top of an Windows Server installation, while the Database server must run a version of Microsoft SQL Server Standard.

Amazon's cloud hosting service is named Elastic Compute Cloud (EC2). In this server there are different types of instances that can be rented. Since MOSS 2007 requires at least 2GB of memory, with the recommended being 4GB[120], it is necessary to use the large instances of EC2 for the Front end Web servers and for the Application server. These instances provide 7.5GB of memory, and 850GB of instance storage in a 64 bit-platform. All these instances should run the Windows Server 2003 machine images provided by Amazon. The Database

server should be also a large instance running Windows Server with SQL Server Standard[121]. So the MOSS farm in AWS will consist in four large instances three of which will be running Windows Server 2003 machine images and the fourth will run Windows Server 2003 with SQL Server Standard.



*Figure 3 Amazon Web Services – Medium Server Farm*

- **Hours of Development:** The average amount of hours necessary to develop a CMS with MOSS 2007 was estimated to be 1600 hours, reflecting the fact that on average a project team is composed by 5 people working during 8 weeks (8 hours a day);
- **Codebase Type:** Independent web applications can be developed on top of the same MOSS 2007 installation, making it a single codebase solution;
- **Security:** MOSS 2007 comes with many security functionalities [122];
- **Ease of Use:** There are different functionalities in MOSS 2007 that make it very easy to use[122],
- **Built-in Applications:** This CMST comes with many different built-in applications which might be useful for the development of some CMS projects that need those applications but can make it a bit more complicated to start other projects, due to the necessity to remove the unwanted applications [122];
- **Added Value:** MOSS as been regularly used by Tam Tam in different types of projects, so the knowledge in the company is quite high and evolving everyday with knowledge sharing sessions and a high level of communication between the developers. So the added value was considered to be high;
- **Deployment:** The installation of a MOSS server farm is not a simple process, but Tam Tam is developing a script to speed up and simplify the process. That script after being done would also work with AWS server farm. After installing the MOSS Farm and setting the necessary environments (test, acceptance and



production) web applications are easily moved from one environment to the other;

- **Update Web Applications:** Updating the Web applications in AWS is quite simple even though it might result in period of downtime. Nevertheless there are services at AWS that avoid any downtime. One of the services that can be used for this is named Elastic IP Addresses, which are IP addresses that are associated with an account at AWS and which can be map or remapped instantly to different instances. So if it is necessary to avoid downtime on some applications a new instance with those applications can be launched in AWS and the Elastic IP Addresses can be mapped to the new instance. Then the old instance can be updated, and after that update process is finished just remap the IP Addresses to the updated instance;
- **Portability:** There is no difference between a MOSS Web application developed to work on the AWS or, for instance, in a TerreMark hosting solution. So it would make a move from the AWS hosting to another solution;
- **Scalability:** With Amazon Web Services each instance is created with an Amazon Machine Image. By using this image it is a simple and fast task to scale up and down the web applications, to meet the demand. Instances can be created from another instances that are already running;
- **Redundancy:** In this scenario there is redundancy in terms of the Front end Web servers, that handle the HTTP requests, but not of the Application Server or the Database;
- **Load Balance:** In this scenario load balancing exists between the two Front end Web Servers, and can be easily done in EC2;
- **Backups of Application and Data:** With Amazon Web Services backups can be easily made, using Amazon EBS service, that was previously presented;
- **Backups Redundancy:** To assure backups redundancy Amazon also has a feature to allow the storage of backups in multiple locations of the same Region (EU or US) where it has its servers;
- **Server Updates:** In Amazon Web Services all the servers can be updated with ease. The process is quite simple, but to ensure that there is no period of downtime the costs will increase slightly. First a new instance per type of server must be launched with a copy of the respective server. The updates should be made there and then it is possible to remap the IP address (through the previously referred Elastic IP Addresses) that is live to the updated instance, dropping the old versions;
- **Server Maintenance:** Server maintenance can be done by using the Amazon Management Console or the Mozilla Firefox extension Elastic Fox. Instances can be easily maintained by Tam Tam systems administrators.
- **Launch Cost:** The launch cost of this scenario consists on the initial investment for buying the software license for MOSS 2007, the Windows Server and SQL

Server licenses are included in the cost of EC2 instances. The instances are charged hourly and based on the usage, so they are not considered part of the launch cost. So considering the four MOSS 2007 for Internet Sites License[123, 124] that are necessary the launch cost was calculated and is €117752.

- **Cost per Project:** In this scenario the cost per project is related with the price of the EC2 instances. These instances are charged hourly and per usage so it was assumed that they would be used every hour of everyday during one year to facilitate the calculations. The total cost of use was divided by the expected average of applications used, that in the actual situation is 45. Although maybe in the first year this scenario would not have that many applications built it was thought to be a simpler way to compare the different scenarios, putting them all in the same context. So the total cost per year is the cost of 3 large EC2 instances running Windows Server with Authentication Services (€0.57 per instance per hour) plus the cost of 1 large EC2 instance running Windows Server with SQL Server Standard and Authentications Services (€0.99 per instance per hour), which represents a total cost per year of €23652. Which divided by 45 projects represents a cost of €525.6 per project per year. This calculations are based in the prices as of June 25th of 2009[121].

#### 5.2.2.2 Scenario 2: Custom CMS + Microsoft's Azure Services Platform

This scenario is composed by a Custom CMS that would be built by Tam Tam, and Microsoft's Azure Services Platform as the hosting solution.

Microsoft has not yet released information about the cost of the services that are provided by Azure Services Platform, but in the beginning of 2009 a top Microsoft executive has stated that their prices are going to be competitive and it would be charged as you go[126]. Having only this information it was assumed for the sake of the comparison that Microsoft's prices would be the same as AWS's prices. Since most of CMST do not require a structure as complex as the one required by MOSS 2007, as presented in the previous scenario, only two servers will be considered for this scenario. One server that will work as the Front end Web server and another one for the Database. In this characterization and specially for the cost calculations it was assumed that the Front end server is the same as an EC2 large instance running Windows Server 2003, and although Azure does not provide relational database services there are some data storage services at Azure nevertheless, so the Database server was assumed to be the same as an EC2 large instance running Windows Server 2003 with SQL Server Standard.

Before going into its characterization it is important to clarify that for this scenario, some information was looked at as part of the tool requirements, since it is the possibility of building the tool that is being analysed and not an actual existing tool. For some of the criteria analysed it was necessary to have an idea of the complexity that the tool should have, so and based on the existing scenarios it was assumed that a tool built by Tam Tam would be simpler than MOSS, since this provides tools to build a wide range of different applications not only CMS. And also slightly simpler than SmartSite (another tool currently being used by Tam Tam), since in

opposition to SmartSite this custom CMS would be built specifically for Tam Tam based on Tam Tam's requirements.

- **Hours of development:** it is estimated that to build a CMS in this scenario it would take 320 hours, consisting of 2 developers working on the project for 4 weeks. This was based in the fact that to build a project with SmartSite it takes in average 400hours. The difference is related with the fact that when Tam Tam builds a project with SmartSite there is an initial phase where it has to remove unwanted functionalities that come with the start kit of SmartSite, and is also related with the difference in complexity assumed for the custom tool in relation to SmartSite;
- **Codebase Type:** Although it was part of the initial requirements of Tam Tam the complexity of developing a tool that allowed for different applications to run on top of the same installation seemed too much and maybe unrealistic in this context. So it was assumed that in a first development the tool would not support that;
- **Security:** This CMST should provide the security functionalities required by Tam Tam to ensure the satisfaction of its clients;
- **Ease of Use:** The development of its own custom tool allows for Tam Tam to define the way it should be used. So this tool should have the basic characteristics that ensure that the resulting applications are as easy to use as possible;
- **Built-in Applications:** In terms of built-in applications this tool should be equipped with the applications that are normally part of CMS projects' requirements to facilitate the development of every new project;
- **Added Value:** A tool that is developed to meet the requirements of Tam Tam can be considered as having a very high added value;
- **Deployment:** Deployment should be as simple and easy as possible for CMS built in this scenario. The development and deployment of applications for Azure Services Platform is made easy with a set of developer tools that are integrated with Microsoft Visual Studio and come with a publish functionality that make it easy to deploy on Microsoft's Cloud. After publishing the applications in the Cloud stay in a staging machine at the Cloud and then should be moved by the developers into the Production machine, and only then they will be operational and accessible to visitors;
- **Update Web Applications:** When updating an Web application hosted in Azure the updated application, as in the deployment, goes to the staging machine and then can be moved to the production machine. The process of moving the application to the production machine causes some downtime of the application;
- **Portability:** Developing an application to run on Microsoft's Azure Services Platform is not exactly the same as developing a traditional ASP .NET application. The Application Programming Interfaces (API) available in the Cloud

are different from the ones available for applications to run on-premises servers. At this moment Azure Services Platform also does not provide relational database services, providing simpler data storage services through Windows Azure [127]. So it would be necessary to make changes to an application if one was to move it from Azure to another type of hosting where relational databases are supported. Nevertheless Microsoft is working on the development of a relational database that will work on its cloud, provided by its SQL Services [128];

- **Scalability:** Windows Azure allows for easy and quick scalability of Web Applications, directly from the management panel;
- **Redundancy:** In the defined scenario with the defined server structure redundancy does not exist. But Azure provides tools to scale up the application so that it becomes redundant;
- **Load Balance:** With only one Front end Web server there is no possibility to use load balancing. Nevertheless the functionality exists in Azure;
- **Backups of Application and Data:** Azure Services Platform provide means to do backups of the information with ease;
- **Backups Redundancy:** As with other Clouds Azure provides ways to do backup redundancy;
- **Server Updates:** It is not expected that any server updates might cause downtime periods in Azure;
- **Server Maintenance:** No server maintenance is required in Azure Services Platform;
- **Launch Cost:** In this scenario there are no software licenses to acquire. The cost of launching this scenario is associated with the cost of developing the CMST by Tam Tam developers. It was estimated that for a project like this one it would be necessary to have two developers working on it during one year and a half. So having 2 developers working the project for 16 months, and receiving an average salary of €3000 it would result in a cost of €102000 to build the CMST and start using it.
- **Cost per Project:** In this scenario the yearly cost per project was calculated in the same way as the previous scenario. In this scenario there are only two instances. One large EC2 instance running Windows Server 2003 with Authentication Services (€0.57) and one large EC2 instance running Windows Server and SQL Server Standard with Authentication Services (€0.99) resulting in a total cost per year of €13665.6. Which, with 45 applications running in the scenario, results in a cost of €303.68 per project per year.

### 5.2.2.3 Scenario 3: Umbraco + TerreMark

The third scenario has Umbraco as the CMST and TerreMark as the hosting solution.

This scenario's server structure is similar to scenario number two and consists of a Front end Web Server and a Database server, since this tool does not need any special server configuration as well.

- **Hours of Development:** Umbraco is a CMST fairly easy to use and Tam Tam has already developed some projects with it. So it was estimated that to build a new project it would be necessary to spend 400hours, consisting of 2 developers working on a project for 5 weeks;
- **Codebase Type:** It is not possible to create different and independent Web Applications with a common codebase, with this CMST.
- **Security:** Umbraco is a CMST that has a good number of security functionalities included, but slightly less than MOSS 2007[39];
- **Ease of Use:** In terms of use this CMST possesses good tools to facilitate the use of the resulting applications[39];
- **Built-in Applications:** There are less built in applications in Umbraco than in MOSS 2007, which can be considered better or worst depending on the situation and on the requirements of the project at hands[39];
- **Added Value:** Tam Tam's developers do not have much experience with this CMST, but since it is built with the .NET Framework, the added value can be considered to be medium;
- **Deployment:** The deployment of an Umbraco Web application in TerreMark servers would be made through the use of a File Transfer Protocol application. The installation of the application is very simple;
- **Update Web Applications:** Updating the Web Applications is not difficult. It is only necessary to replace the existing files with the latest binaries. This might cause just a few moments of downtime;
- **Portability:** An Umbraco Web Application is easily portable from TerreMark to Amazon Web Services or other hosting facilities that run .NET Applications.
- **Scalability:** Scaling an hosting solution at TerreMark takes some time because new servers need to be ordered and then installed, taking up to 15 days;
- **Redundancy:** Redundancy is not provided with this solution.
- **Load Balance:** Since there is only one Front end Web server there is no load balancing;
- **Backups of Application and Data:** TerreMark makes daily backups of the system and contents.
- **Backups Redundancy:** These backups stay in TerreMark's premises only. So there is no backup redundancy.
- **Server Updates:** Any server update will result in server downtime.

- **Server Maintenance:** Most server maintenance is done by Tam Tam's employees, through Remote Desktop. The server maintenance tasks are all of the responsibility of Tam Tam;
- **Launch Cost:** The launch cost in this scenario was defined as the price of the Hardware of the servers, which has the price of the necessary licenses included and is €3000 per server. Which results in a total of €6000.
- **Cost per Project:** In terms of cost per project it was considered the monthly cost of having the servers running on TerreMark's premises. This cost is of €600 per server. So two servers working during one year result in a cost of €14400. Which gives a cost of €320 per project per year.

### 5.2.3 Scenarios Variations

After the characterization of the three selected scenarios it was noticed that in some criteria the scenarios did not perform very well. This was most visible in terms of the hosting characteristics. This situation happened due to the choice of services and server structure that were made during the scenario definition. In most of the cases the hosting providers had other services that could improve the performance of the scenarios. So, although some of these alternatives could carry an increase in the costs associated with the scenarios, it was decided to expand the set of scenarios. The aim was to take advantage of the alternative services available.

The new scenarios that were added are based in the first three scenarios. One new scenario was added per each of the existing ones. These new scenarios share some of their characteristics, especially the ones connected with the CMST. In the characterization of these scenarios only the characteristics that are different from the scenario in which they were based will be listed.

#### 5.2.3.1 Scenario 1.2

Scenario 1.2 is a variation of Scenario number 1. This scenario varied only in terms of the server structure assumed. It was assumed for this scenario a server structure with five servers instead of the four servers from scenario one. The fifth server is a second Database server that will provide redundancy of the content of the different applications. For this scenario only the information related to Redundancy, Load Balance, and Cost per Project criteria is different from the scenario in which it is based.

- **Redundancy:** In this scenario two instances are used both for the Front end Web servers and for the Database servers. Assuring redundancy in the Web Server, and in the Database Server.
- **Load Balance:** The use of two instances for the Web Server allows for Load Balance, also the existence of two instances for the Database it is possible to balance the load of the requests to the database;
- **Cost per Project:** In this scenario the cost per project like in scenario 1 is related

with the price of the EC2 instances. These instances are charged hourly and per usage so it was assumed that they would be used every hour of everyday during one year to facilitate the calculations. The total cost of use was divided by the expected average of applications used, that in the actual situation is 45. Although maybe in the first year this scenario would not have that many applications built it was thought to be a simpler way to compare the different scenarios, putting them all in the same context. So the total cost per year is the cost of 3 large EC2 instances running Windows Server with Authentication Services (€0.57 per instance per hour) plus the cost of 2 large EC2 instance running Windows Server with SQL Server Standard and Authentications Services (€0.99 per instance per hour),, which represents a total cost per year of € 32324.4. Which divided by 45 projects represents a cost of €718.32 per project per year. This calculations are based in the prices as of June 25<sup>th</sup> of 2009[129].

#### 5.2.3.2 Scenario 2.2

This scenario was obtained by imposing a variation in the server structure of the scenario number 2. This variation consisted on the increase of the number of servers from 2 to 3, with the addition of a second Front end Web server.

- **Redundancy:** In this scenario there is redundancy in the Front end servers, since there are two instances allocated for this;
- **Load Balance:** The use of two instances for the Web Server allows for Load Balance. Avoiding any of the Web Servers to be overloaded and reduce the quality of the service.
- **Cost per Project:** In this scenario the yearly cost per project as in the scenario 2 but in this there is one extra instance, namely one more large EC2 instance running Windows Server 2003. So there are in total 2 large EC2 instance running Windows Server 2003 with Authentication Services (€0.57) and 1 large EC2 instance running Windows Server and SQL Server Standard with Authentication Services (€0.99) resulting in a total cost per year of €18658.8. Which, with 45 applications running in the scenario, results in a cost of €414.64 per project per year.

#### 5.2.3.3 Scenario 3.2

This scenario is derived from the scenario composed by Umbraco and TerreMark hosting solution. In this case not only the server structure is changed but also the hosting provider. So instead of TerreMark the hosting provider for this scenario is Amazon Web Services. This variation was made not only to improve the characteristics associated with the server structure but also because the Scenario number 3 presented the lowest costs and it looked interesting to analyse how would an Open Source CMST like Umbraco do, in terms of the defined criteria, with a Cloud hosting solution.

In terms of server structure was assumed a structure with two Front end Web Servers for redundancy and load balancing and a Database server, which in Amazon represents two large instances of EC2 service running Windows Server and one large instance of EC2 running Windows Server with SQL Server Standard.

The criteria that suffered a change with this variation are the following:

- **Deployment Process:** The deployment of an Umbraco Web Application in Amazon Web Services instances would be made through the use of Amazons Management Console or Elastic Fox, an extension for Mozilla Firefox. After deployment of the system it needs to be installed, which can be done by using the same tools.
- **Scalability:** With AWS each instance is created with an Amazon Machine Image. By using this image it is a simple and fast task to scale up and down the web applications, to meet the demand. Instances can be created from another instances that are already running;;
- **Redundancy:** Redundancy exists in terms of the Web server, but not in terms of the Database server;.
- **Backups of Application and Data:** With Amazon Web Services backups can be easily made, using Amazon EBS service, that was previously presented;
- **Backups Redundancy:** To assure backups redundancy Amazon also has a feature to allow the storage of backups in multiple locations of the same Region (EU or US) where it has its servers;
- **Server Updates:** In Amazon Web Services all the servers can be updated with ease. The process is quite simple, but to ensure that there is no period of downtime the costs will increase slightly. First a new instance per type of server must be launched with a copy of the respective server. The updates should be made there and then it is possible to remap the IP address (through the previously referred Elastic IP Addresses) that is live to the updated instance, dropping the old versions;
- **Server Maintenance:** Server maintenance can be done by using the Amazon Management Console or the Mozilla Firefox extension Elastic Fox. Instances can be easily maintained by Tam Tam systems administrators.
- **Launch Cost:** Since in this scenario the software licenses are included in the monthly price of the instances used the launch cost associated with this scenario is €0.
- **Cost per Project:** In this scenario the cost per project is related with the price of the EC2 instances. These instances are charged hourly and per usage so it was assumed that they would be used every hour of everyday during one year to facilitate the calculations. The total cost of use was divided by the expected average of applications used, that in the actual situation is 45. Although maybe in the first year this scenario would not have that many applications built it was



thought to be a simpler way to compare the different scenarios, putting them all in the same context. So the total cost per year is the cost of 2 large EC2 instances running Windows Server with Authentication Services (€0.57 per instance per hour) plus the cost of 1 large EC2 instance running Windows Server with SQL Server Standard and Authentications Services (€0.99 per instance per hour),, which represents a total cost per year of €18658.8. Which divided by 45 projects represents a cost of €414.64 per project per year. This calculations are based in the prices of EC2 instances as of June 25<sup>th</sup> of 2009[129].

## **5.3 Summary**

In this chapter was described the process of defining and characterizing the alternative scenarios for Tam Tam's CMS Projects.

The chapter starts with the description of the research and definition of the alternative scenarios. The different choices made during this steps and the different scenarios are described.

This chapter also convers the definition of the criteria that were used to do the characterization of the scenarios. After the criteria have been described the chapter focus is put on the characterization of the scenarios.

The chapter ends with the definition and characterization of some variations of the previously defined scenarios. These variations gave origin to three new scenarios that were thought to be good additions to the set of alternatives.

In the end of the chapter six scenarios have been defined and characterized. These six scenarios will be evaluated and compared in the Chapter 7 of this report.



## Chapter 6

# MCDA Choice and Implementation

*Honor is not about making the right choices. It is about dealing with the consequences.*

Midori Koto [130]

This chapter describes the choice of the MCDA methodology and the process of building an Application that implemented the chosen methodology.

### 6.1 Methodology Selection

Choosing a MCDA methodology is not a simple task. There are many different methodologies available and there is not a strict framework regarding the field applicability of each method. The first step towards the choice of a methodology was to define a small set of methodologies to choose from. This set was compiled from the three main types of MCDA methodologies and included the SMART methodology, AHP, and ELECTRE. The objective was to verify if some and which of these methodologies would make a good (better) way to help Tam Tam. In case none of these would seem good enough, more methodologies could be brought in for appreciation.

SMART was the first methodology analysed and it was also the first to be discarded. This methodology is very simple and straightforward, with the weights of the criteria and of the alternatives being attributed directly. This way of defining the weights did not seem very easy to work with, especially when the DMs are not familiar with MCDA methodologies.

The ELECTRE methodologies were the next to be analysed. These methodologies are said to give more control to the DMs when it comes to deciding the importance of the available options. This is done by defining some thresholds for concordance (agreeing) and discordance (disagreeing). These thresholds are then used to analyse the options that were previously compared. Although the DM might have some control over the decision it is not entirely clear how should these thresholds be defined, and so the decision might look somewhat arbitrary.

Like in the case of SMART, the fact that the DM has to define or arbitrate some values is not very positive especially when it he/she is not familiar with this type of methodologies, and when there is not enough time to learn complicated concepts that would require practice to handle.

Given the shortcomings of these two alternatives we set out to analyse AHP. One of the first aspects that caught our eye when we look into AHP in depth was the amount of controversy surrounding this methodology. One of the major criticisms made to AHP is that this methodology is very vulnerable to rank reversal[131], but other concerns have also been raised, some of which related to the axioms on which AHP is based. Lewis Warren[132] defines it as an ad-hoc set of axioms. Other authors argued that the problems assigned to AHP are not critical and can be overcome, supporting that the AHP is suitable for decision aiding [133, 134]. Some studies were found that compared AHP with other methodologies but have not been conclusive in relation to AHP's problems in practical applications[135]. Despite the existing criticisms to this methodology is still one of the most widely used in confronting and addressing practical problems[136, 137].

Having analysed the different methodologies and pondered the pros and cons of the various alternatives it was decided to use AHP to help Tam Tam analyse the CMS Projects' problem. This decision was made with the notion that some caution had nevertheless to be taken to avoid the known problems referred in the literature surveyed..

The bases for this decision were the easiness of judgement assessment, its practicality demonstrated by its application to different real life problems, and also the wish to experiment ourselves the application of AHP to a real problem. Since this methodology was taught at the "Faculdade de Engenharia da Universidade do Porto", in the course of "Decision Support Systems" of the Integrated Master in Informatics Engineering and Computation", it was also important that the methodology was already previously known. This familiarity could be important when it was time to explain the methodology to the DMs of Tam Tam.

## **6.2 Implementing the AHP Methodology**

A small presentation of AHP methodology was made at Tam Tam with a demonstration of use with an Microsoft Excel sheet. After the small presentation and demonstration, the use of the methodology was approved, although the use of Excel sheets seemed a bit discouraging. The number of tables needed to cover all the pair wise comparisons was too high and the overall perception was that it was too complex to use. This was one of the reasons that led to the development of a software application that implemented AHP. This decision was made although there are already a few tools that implement the AHP methodology, some of which are free, Web based and quite simple to use[138].

The reasoning behind this decision was the fact that building a flexible application could not only help Tam Tam with the main problem of this project but also with other problems that it might face in the future, and which might need a thorough analysis and the use of a MCDA methodology. Implementing a tool from scratch also helped to analyse the problem further more and also to understand with more detail the AHP methodology.

Since Tam Tam is a Web development company it was decided that the tool should be Web based so that, after being finished, it could be seen as a possible addition to other products

made by Tam Tam, in case some client requires some decision aiding tool for their applications. Being a Web application also allows it to be simultaneously used by the different Decision Makers, involved in a decision making process, without the need to install a new software.

Beside these factors, the implementation of a tool like this was also seen as a challenge worth taking.

## **6.2.1 Requirements Specification**

Before the implementation of the application a set of requirements was defined to support the application. These requirements are presented in the next sub sections.

### **6.2.1.1 Application Perspective**

The application should work independently of any other application and will be self-contained, still the possibility of it being integrated with other applications in the future is very much present. The application should be built to run on a Web server so that it can be accessed by different users at the same time through the Internet. This way it will be possible for a group of Decision Makers to access the application at the same time and independently to one another so that afterwards they can compare and analyse the results obtained by each one of them.

### **6.2.1.2 User Interfaces**

This Web application should have just one user interface, where the users would be able to access its functionalities.

### **6.2.1.3 Software Interfaces**

To access and use this Web application the user must have Internet access and a Web browser program (like for instance Internet Explorer, Mozilla Firefox, Opera, etc.).

### **6.2.1.4 Communication Interfaces**

The only communication interface is the Internet, under the TCP/IP protocol.

### **6.2.1.5 Application Functionalities**

The application should have functions to allow the users to:

- Start a new decision analysis process;
- Input the existing criteria for the specific problem in analysis;
- Input the existing scenarios for the specific problem in analysis;
- Executed the different steps of the AHP methodology;
- Do a sensitivity analysis of the results obtained with the execution of the AHP

methodology;

- Save the state of the process;
- Load a previously saved decision analysis process;
- Navigate back and forth from any point of the decision analysis process;

These functionalities are depicted in the Use Cases diagram of Figure 4.



*Figure 4 Web Application Use Cases*

The system is intended to be easy-to-use, to have the necessary information to guide the user through the whole process, and to be fast to use.

#### **6.2.1.6 User Characteristics**

This Web application should not have user access levels. All the functionalities should be available for all who access the application. Nevertheless it should be taken into consideration that the application will be used both by MCDA literates and by Decision Makers who might not be familiar with the AHP methodologies, so its use should be simple and clear for a first time user.

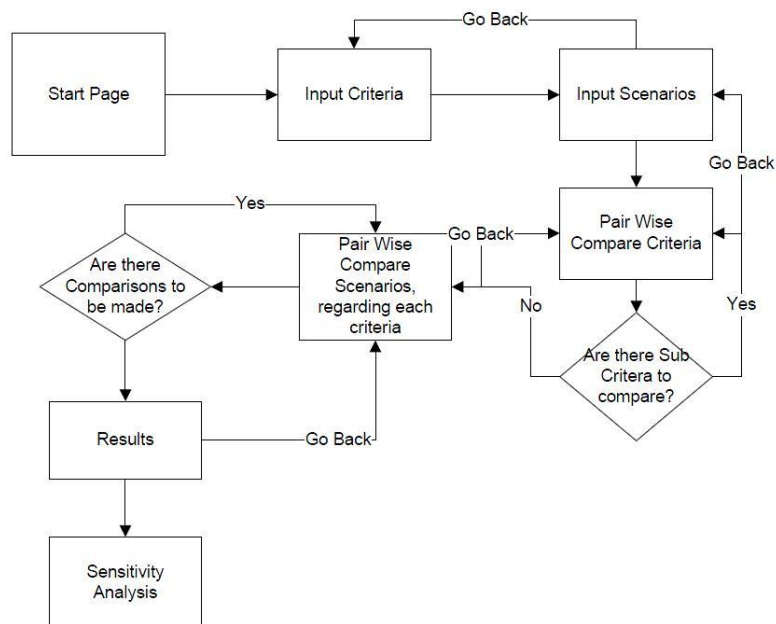
### 6.2.1.7 Constraints

It was defined that the tool should be developed with Microsoft Silverlight [139]. This technology is still quite recent and it seems very appropriate to develop a Web application that will implement a methodology like AHP, which needs the user to fill different fields that are unknown at compile time. Although there are other alternatives to Silverlight, this tool is built in the .NET framework and is a product of Microsoft, so and since Tam Tam is a Microsoft Gold Partner, the use of Microsoft's technologies is encouraged.

### 6.2.1.8 Sequence of Use

In this application there will be only one type of user that will be able to execute all the functionalities defined above in Section 6.2.1.5 . So the sequence of use of the application should be as simple as possible (the sequence of use is depicted in Figure 5).

The user should access the application through its Web browser by pointing to the address where the application is hosted. Then he should have the option of starting a decision aiding process from the beginning or loading a previously saved configuration. If he decides to start from the beginning, he should be guided to the first screen of the process which should have information explaining which type of information he must provide and what must he do to proceed to the next step. This process should be sequential, like the AHP methodology process is, and every new screen should provide information on how to get to the next one. In every screen there should be the possibility for the user to download a file with the state of the process up to that point to be used in the future. There should also be the possibility for the user to move back to a previous step.



*Figure 5 Sequence of Use*

If the user chooses to load a previously saved configuration it should be prompted to choose a file and the application should then analyse the file to verify if it is a valid file for

running the application. The application should also verify in which stage was the process interrupted and allow the user to move to the last stage or to a previous stage. From there the user should be able to continue the decision aiding process.

In the end of the process the results of the application of the AHP methodology should be presented to the user as well as the possibility to do a sensitivity analysis of the process and of the results. From the last screen the user should also be able go back to the previous screen.

In the last screen the user should be able to download a file with all the information related with that decision aiding process.

## 6.3 Implementation Details

After having the requirements specification it was time to move on to the implementation of the application. The first step of the implementation was to define how would the application implement the AHP methodology, how would the different steps of the methodology be implemented so as to comply with the sequence of use presented in the requirements specification, and how would the information be handled by the application.

Next will be presented how the AHP process was implemented, and after that will be explained how the information is handled in the application. After that, some information related with other functionalities will be presented as well as an analysis of the current state of the application. The application has not reached its final version, with some details still needing to be looked at.

### 6.3.1 Implementation of the AHP Process

The AHP Process starts with the definition of the goal of the analysis, the set of criteria, the set of alternatives, and their characteristics regarding each criterion. After having defined all of this it starts the phase of the pair wise comparisons, first between the criteria, then between each sub criteria that share the same parent criteria, and then the pair wise comparison between the scenarios regarding each of the criteria and sub criteria. At last the results of the comparison are presented and the priority of each scenario is calculated using the different comparisons.

To implement this process into a Web Application it was decided that there would be a page per step, it was also decided that the application would be as flexible and dynamic as possible so there are no limitations regarding the number of criteria and scenarios that the user can create. The pages created to implement the decision aiding process, their characteristics, and some of the available functionalities will now be detailed:

- 1- **Criteria Page:** This page was built with a form where the user can fill the information about the defined criteria. The information that it is needed for each criterion is its name, its type and a short description that describes it. There is also the possibility of defining a parent for a criterion, this is done by introducing the id number of the parent criterion. A criterion that has sub criteria should have the type “Parent” associated with it. The form starts with space for only two criteria but more can be added easily by click in “Add More Criteria”, extra criteria fields can be removed by clicking in “Remove Extra Criteria”. After filling the form the user can advance to the next step,



by clicking on the button labelled “Next Step”. In Figure 6 it is possible to see the first page of the AHP process with place for six criterion, and with five criterion filled, one of each is a parent of two other (the “Cost” criterion” is parent of “Launch Cost” and “Per Project Cost”);

|   | Criterion Name     | Data Type  | Short Description                           | Parent Criterion |
|---|--------------------|------------|---|------------------|
| 1 | Resources          | Integer    | The resources necessary for the project.    |                  |
| 2 | Redundancy         | Binary     | Does the hosting provide redundancy         |                  |
| 3 | Cost               | Parent     | General Criteria for Cost                   |                  |
| 4 | Launch Cost        | Monetary   | The cost of launching the solution          | 3                |
| 5 | Per Project Cost   | Monetary   | The cost associated with each independent   | 3                |
| 6 | Application Update | Complexity | How complex it is to update the Application |                  |
| 7 | Enter a name...    |            | Enter a description...                      |                  |

*Figure 6 Page of the first step of the AHP decision aid process – Defining the criteria*

- 2- **Scenarios Page:** After filling the information about the criteria it is time to fill the information related with the different alternatives, here named scenarios. In this page it is also possible to increase the number of scenarios to be considered. Here the user should fill a short description for each of the scenarios and then fill up the information of each scenario for all the criteria that have been defined in the previous page. After having filled in the scenarios that it wanted to fill the user should click in the “Next Step” to go to the start of the pair wise comparisons. Figure 7 shows the scenarios page. In this example the user is filling three scenarios and has already filled in the information related with the previously defined criteria.
- 3- **Comparing Criteria:** After having defined the criteria and the different scenarios, the user is presented with this page where he has to do the pair wise comparisons of the different criteria. As one can see in Figure 8, these comparisons are made by choosing the value from a drop-down list that represents the user’s feelings when comparing two different criteria. The scale used is the scale proposed by Saaty [117] for the pair wise comparisons with AHP, which was presented in sub section 3.2.2. With this application it is only necessary to do  $n*(n-1)/2$  comparisons each time, with  $n$  being the number of criteria, because whenever a comparison is made the opposite value is immediately updated. This page also includes a chart that represents with a line series the values of each criterion relative to the other criteria to which it is being compared. The priorities

of the different criteria are calculated on the run each time a value is changed and are represented in the last column of the table. The application calculates the priorities of each row based on a formula that is an alternative to Saaty's method[117]. The method propose by Saaty consists in calculating the “weights [of each row] as the elements in the eigenvector associated with the maximum eigenvalue of the matrix.”[108] while the other methodology is easier to apply and consists in: calculating the geometric mean of each row in the matrix; calculating the total of geometric means; and normalising each row's geometric mean by dividing it by the total of the geometric means[108]. If any sub criteria were defined clicking in the “Next Step” button will take the user to a page similar to this one where the user will have to make the pair wise comparison between the sub criteria. After all the pair wise comparisons of criteria and sub criteria are done clicking on the “Next Step” button will take the user to the scenarios' comparison page;

|                           | Scenario 1        | Scenario 2  | Scenario 3          |
|---------------------------|-------------------|-------------|---------------------|
| <b>Short Description</b>  | Custom CMS in Azu | MOSS in AWS | Ibraco in TerreMark |
| <b>Resources</b>          | 2                 | 5           | 2                   |
| <b>Redundancy</b>         | No                | Yes         | No                  |
| <b>Cost:</b>              |                   |             |                     |
| <b>Launch Cost</b>        | 128000            | 60000       | 6000                |
| <b>Per Project Cost</b>   | 430               | 400         | 375                 |
| <b>Application Update</b> | Simple            | Average     | Simple              |

*Figure 7 Page of the second step of the AHP decision aid process – Defining the Scenarios*

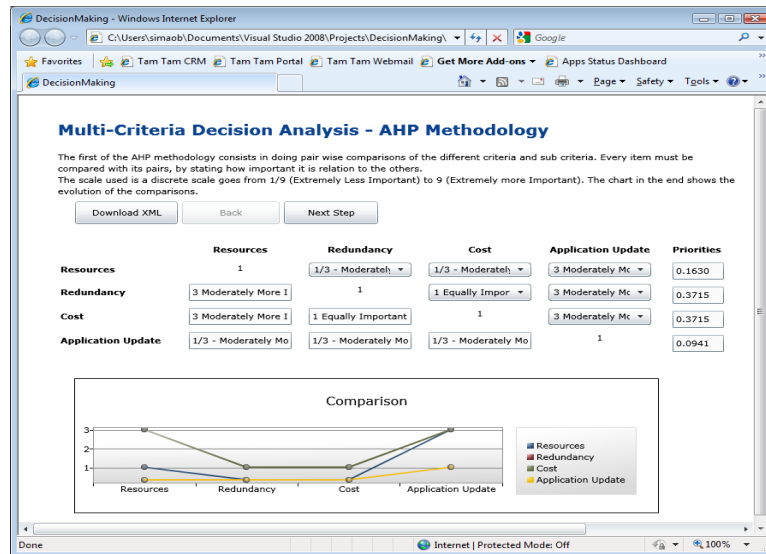


Figure 8 Page of the third step of the AHP decision aiding process- Pair wise comparison of the criteria

- 4- **Scenario Comparison:** This page's structure is similar to the previous one, but in this one the targets of the comparison are the defined scenarios with regard to a specific criterion. After each comparison is made and after the user clicks in the "Next Step" button it will be taken to the next comparison regarding another criteria. To help the user make the comparisons this page displays the values of each scenario for the criteria that is in focus. An example of this page can be seen in Figure 9. After all the comparisons are made the user will be taken to the results page.

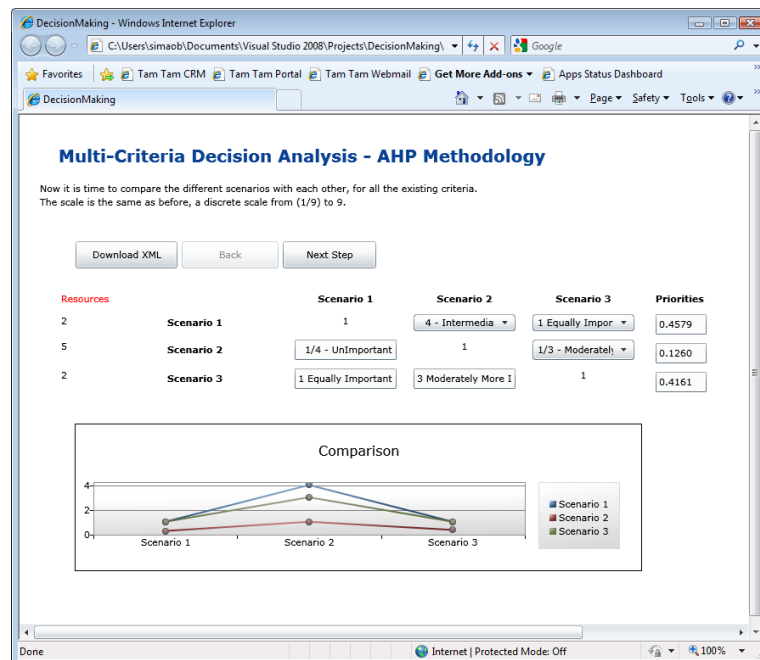


Figure 9 Page of the fourth step of the AHP decision aiding process- Pair wise comparison of the scenarios in relation to the each criteria

- 5- **Results Page:** The last page of the AHP decision aiding process is where the results of the application of the methodology are presented. As it can be seen below in Figure 10 this page lists the different criteria and sub criteria and ranks them in relation to their global priorities. In terms of the main criteria their global priorities are the priorities defined by the user through the pair wise comparison, while in terms of sub criteria the global criteria are calculated by multiplying the user defined priorities (which represents its local priority) by the sub criteria's parent priority. A table is presented below the criteria list that displays the results of the scenarios' pair wise comparison already multiplied by each criteria. The highest result per criteria is presented in bold. In the right side of the table is the total priority of each scenario. In this example one can see that the recommended scenario would be scenario number one. A chart is placed on top of the table on the right side with a series per scenario and an extra series with the maximum value in each criteria. This chart is a visual representation of what can be seen in the table and it aims at helping the user interpret the results. Besides the various results this page also presents a functionality to perform a sensitivity analysis over the results of the AHP methodology. The sensitivity analysis functionality is still not completely finished. Nevertheless it already makes a simple analysis, by making slight changes to the priorities of the criteria and re-calculating the results of the process, with those changes.

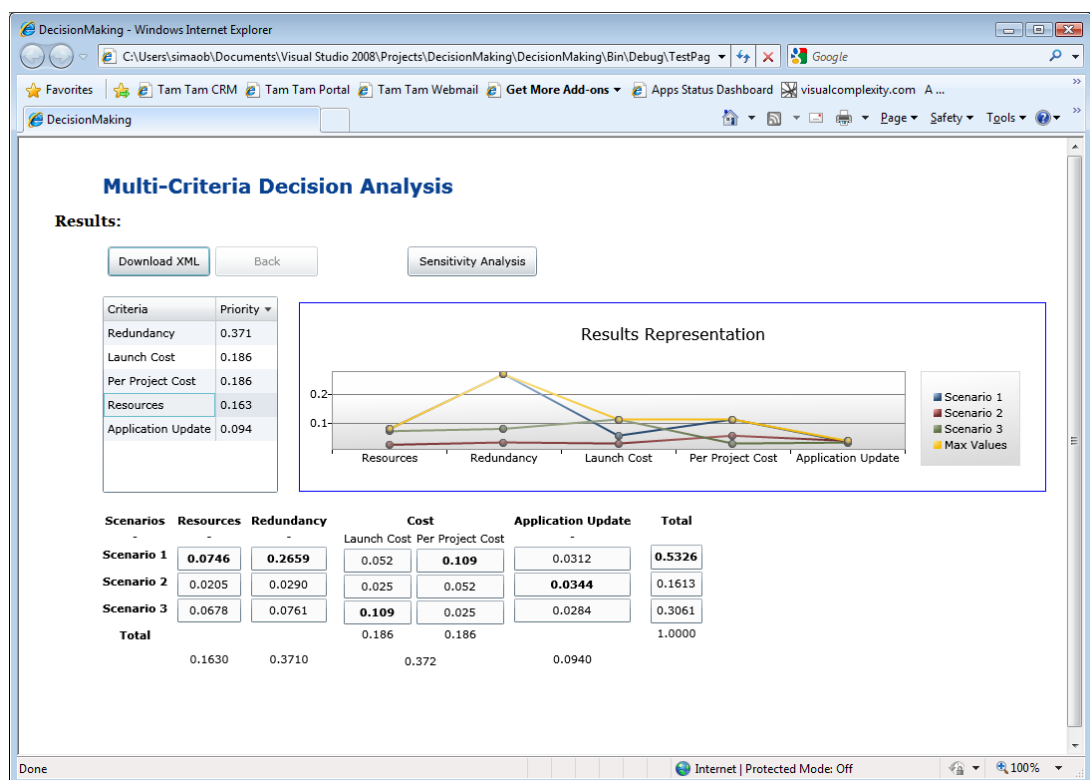


Figure 10 Results Page of the AHP decision aiding process

The sequence of use is illustrated in Figure 5, above.

### 6.3.2 Handling the Information

Before developing the application there were two concerns regarding the information necessary for the execution of the AHP methodology: storage and information handling.

Regarding the storage of the information it did not seem necessary to use a full relational database to support this application, because each use of the application is independent from other uses, and there are no access levels that require a database. Nevertheless it was part of the requirements that the user should be able to store the state of the execution of the process, so there was a need to define a way to store that state. The way the information should be stored is where both the handling and the storage concerns meet. Because it also seemed important that the stored state should be human readable, allowing it to be consulted and to serve as a representation of a particular process outside of the application.

With these ideas in mind it was decided that the information should be stored in XML files, in a readable structure that allowed for any person to identify the different types of information inserted during the execution of the application.

One thing that also contributed to the choice of XML files was the fact that Microsoft Silverlight has some libraries with functions that allow for the easy handling of XML files, namely the System.Xml.Linq. With this library the task of saving and loading saved configurations was made easy. An example of a saved configuration can be seen in Figure 11.

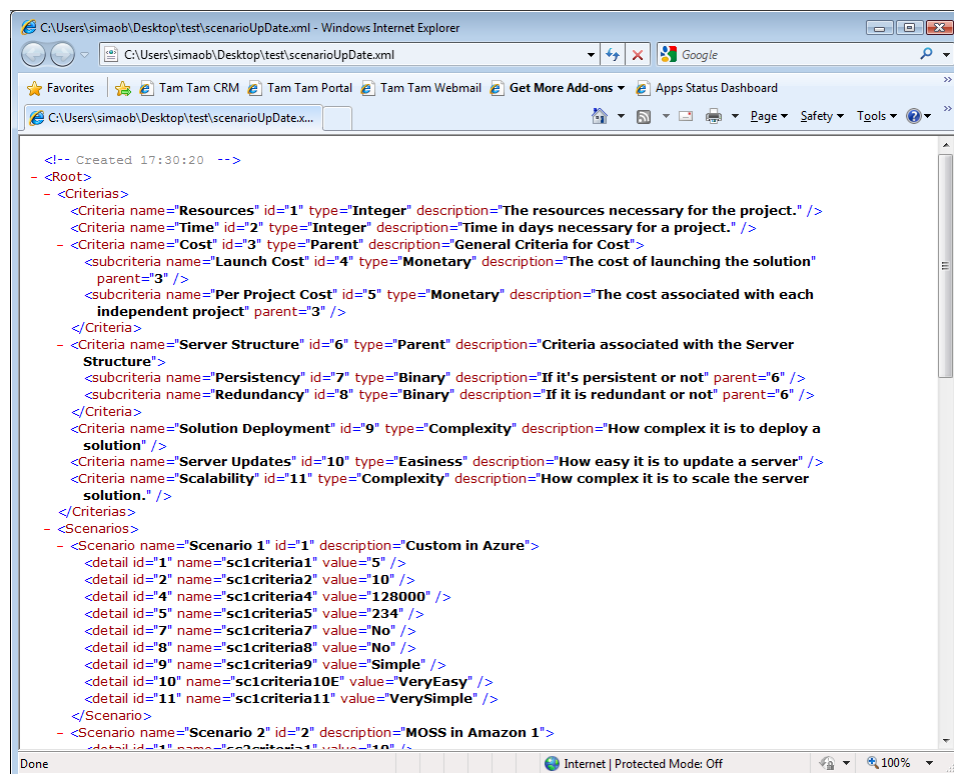
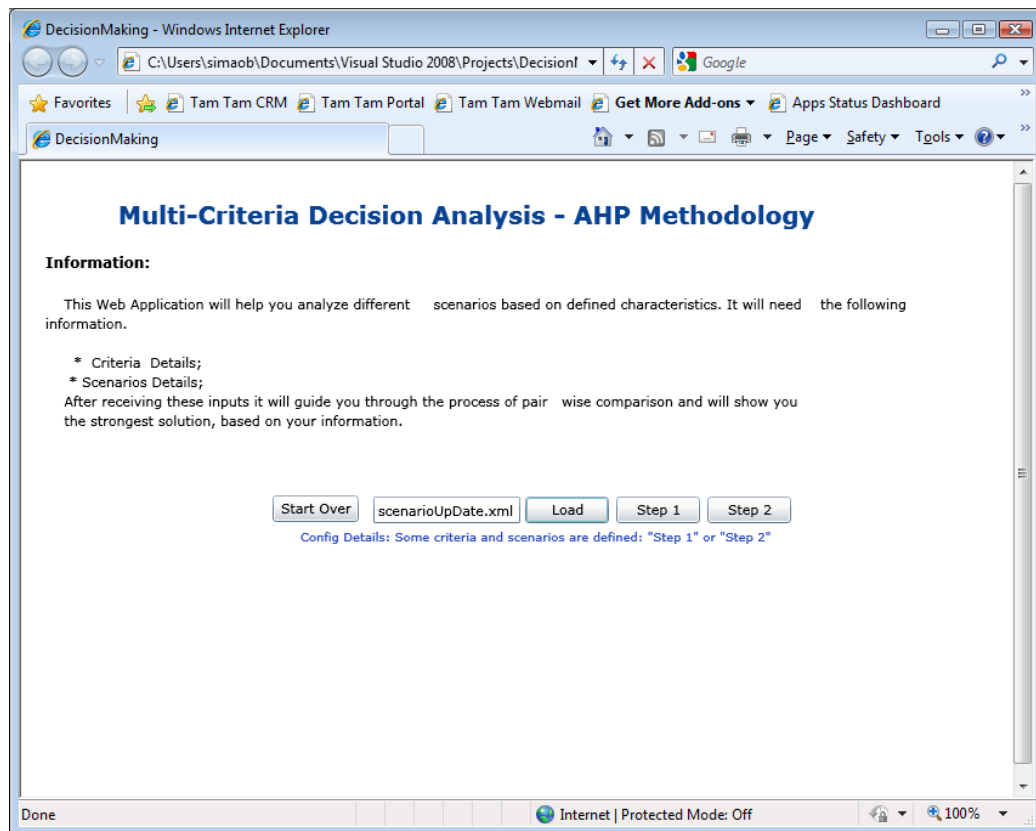


Figure 11 Example of a saved configuration file

The developed applications allow the user to save the state of the process in any of its steps. The loading of a saved configuration can be done from the start page, as it can be seen in Figure 8, and allows the user to advance to the criteria definition page or the scenario definition

page, based on the contents of the file. If the file has criteria defined the user can move to the criteria definition page and edit the existing criteria and continue from there. If the user loads a file with both criteria and scenarios defined it can choose whether to go to the page of the criteria definition or to the page of the scenarios definition.



*Figure 12 Loading a saved configuration*

### 6.3.3 Application Overview

Although the application is not completely finished, the most important functionalities have already been developed and are functioning properly. From the functions specified in the requirements, the sensitivity analysis is the one that needs to be improved the most.

During the development of the application there was the concern of making it as usable as possible. For the sake of usability, the AHP process was built in a way that seemed most simple, and field validation was implemented to help the user avoid input mistakes (which can be seen in Figure 13). Nevertheless there was not enough time to do the necessary usability tests. The only tests done were to try the application with Tam Tam's problem, which will be described in Chapter 7. Although these tests were not real usability tests they in some way helped in the development of the application, and in noticing some of the flaws and things that could be improved.

**Multi-Criteria Decision Analysis - AHP Methodology**

Now please fill the information about the scenarios. Action failed.

Download XML Back Next Step Add More Remove Extra

|                                | Scenario 1  | Scenario 2    | Scenario 3      | Scenario 4        | Scenario 5     |
|--------------------------------|-------------|---------------|-----------------|-------------------|----------------|
| <b>Short Description</b>       | MOSS in AWS | MOSS in AWS.2 | Custom in Azure | Custom in Azure.2 | Umbraco in Ter |
| <b>CMST Related:</b>           |             |               |                 |                   |                |
| <b>HOURS OF Development</b>    | Asdasd      |               | 320             | 320               | 400            |
| <b>Codebase type</b>           | Yes         | Yes           | No              | No                | No             |
| <b>Security</b>                | 10          |               | 10              | 10                | 8              |
| <b>Built-in Applications</b>   | asd         | 8             | asd             | 6                 | 7              |
| <b>Ease of Use</b>             | 9           | 9             | 10              | 10                | 8              |
| <b>Deployment</b>              | Average     | Average       | Simple          | Simple            | Simple         |
| <b>Update Web App.</b>         | Simple      | Simple        | Simple          | Simple            | Simple         |
| <b>Added Value</b>             | 9           | 9             | 10              | 10                | 8              |
| <b>Hosting Related:</b>        |             |               |                 |                   |                |
| <b>Portability</b>             | VeryEasy    | VeryEasy      | Hard            | Hard              | VeryEasy       |
| <b>Scalability</b>             | 0           | 0             | 0               | 0                 | 15             |
| <b>Load Balance</b>            | 1           | 2             | 0               | 1                 | 0              |
| <b>Backup or Data and Annc</b> | 1           | 1             | 2               | 2                 | 2              |
| <b>Backups Redundancy</b>      | 1           | 1             | 2               | 2                 | 0              |
| <b>Server Updates</b>          | 1           | 1             | 2               | 2                 | 0              |
| <b>Server Maintenance</b>      | Simple      | Simple        | VerySimple      | VerySimple        | Average        |

Figure 13 Field Validation (Red fields)

### 6.3.4 Summary

This chapter starts with details of the choice of the MCDA methodology that was used to help Tam Tam solve the CMS Projects problem. The analysis of some MCDA methodologies led to the choice of AHP as the methodology to be used.

After the choice process it is detailed the process of developing an Application that implemented AHP. The decisions made regarding the application and the reasons why it was chosen to build as a Web Application, are also detailed.

In this chapter are also specified the list of requirements for the Web application that implemented the AHP methodology.

After the presentation of the application requirements some implementation details were described. This description covered how the application implemented the AHP methodology, and how was the information handled within the application.

The chapter concludes with an overview of the application and an analysis of what functionalities were implemented at the time of writing of this report..





## Chapter 7

# MCDA Applied to the Problem

*There does not exist a category of science to which one can give the name applied science. There are sciences and the applications of science, bound together as the fruit of the tree which bears it.*

Louis Pasteur [140]

This chapter describes the execution of the developed Web Application to apply the AHP methodology to Tam Tam's Problem. The steps taken before and during the use of the application are described. The execution of the application was done in a meeting with Tam Tam Decision Makers. An analysis of the obtained results is also described in the end of the chapter.

### 7.1 Applying the MCDA Methodology

After the characterization of the scenarios and implementing the Web Application that implements the AHP methodology it was time to apply this methodology to the problem at hands. The methodology was applied together with Tam Tam's Decision Makers.

Before applying the AHP methodology some preparation was required to facilitate the interpretation of the information gathered and to make the use of AHP as clear and as simple as possible.

#### 7.1.1 Information Preparation

To facilitate the consult of the details of all the scenarios, these were compiled into a table. To pass the information of the characterizations into a table a scale had to be used for each criteria. Some of the criteria have a natural scale associated with them but for other criteria it was necessary to define a scale and to convert the information from the characterization into that scale. The scale used for each criteria is presented in Table 2.

*Table 2 Criteria Scales*

| <i>Criteria</i>        | <i>Scale Used</i>                                     |
|------------------------|---|
| Hours of development   | Number of Hours (less is better)                      |
| Codebase type          | (single, multiple)                                    |
| Security               | [0, 10] (10 is better)                                |
| Ease of use            | [0, 10] (10 is better)                                |
| Built-in apps.         | [0, 10] (10 is better)                                |
| Added Value            | (very low, low, average, high, very high)             |
| Deployment             | (very simple, simple, average, complex, very complex) |
| Update Web App         | (very simple, simple, average, complex, very complex) |
| Portability            | (very easy, easy, average, hard, very hard)           |
| Scalability            | Number of Days (less is better)                       |
| Redundancy             | (none, partial, total)                                |
| Load Balance           | (none, partial, total)                                |
| Backup of App and Data | (inexistent, existent, extra service)                 |
| Backups Redundancy     | (inexistent, existent, extra service)                 |
| Server Updates         | (yes; no; no downtime, extra cost)                    |
| Server Maintenance     | (very simple, simple, average, complex, very complex) |
| Launch cost            | Cost associated (less is better)                      |
| Cost per Project       | Cost associated (less is better)                      |

In Table 3 is the information of the different scenarios in relation to the criteria, but presented in each criteria's defined scale.

*Table 3 Scenarios Information*

| <i>Criterion</i>               | <i>Scenario 1</i> | <i>Scenario 1.2</i> | <i>Scenario 2</i> | <i>Scenario 2.2</i> | <i>Scenario 3</i> | <i>Scenario 3.2</i> |
|--------------------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|
| <b>Hours of Development</b>    | 1600              | 1600                | 320               | 320                 | 400               | 400                 |
| <b>Codebase</b>                | single            | single              | multiple          | multiple            | multiple          | multiple            |
| <b>Security</b>                | 10                | 10                  | 10                | 10                  | 8                 | 8                   |
| <b>Ease of use</b>             | 9                 | 9                   | 10                | 10                  | 8                 | 8                   |
| <b>Built-in Apps</b>           | 8                 | 8                   | 6                 | 6                   | 7                 | 7                   |
| <b>Added Value</b>             | 9                 | 9                   | 10                | 10                  | 8                 | 8                   |
| <b>Deployment</b>              | average           | average             | simple            | simple              | simple            | very simple         |
| <b>Update Web App</b>          | simple            | simple              | simple            | simple              | simple            | very simple         |
| <b>Portability</b>             | very easy         | very easy           | hard              | hard                | very easy         | very easy           |
| <b>Scalability</b>             | 0                 | 0                   | 0                 | 0                   | 15                | 0                   |
| <b>Redundancy</b>              | partial           | total               | none              | partial             | none              | partial             |
| <b>Load Balance</b>            | partial           | total               | none              | partial             | none              | partial             |
| <b>Back Up of App and Data</b> | extra service     | extra service       | existent          | existent            | existent          | extra service       |
| <b>Backups</b>                 | extra             | extra               | existent          | existent            | inexistent        | extra               |

| <i>Criterion</i>          | <i>Scenario 1</i>       | <i>Scenario 1.2</i>    | <i>Scenario 2</i> | <i>Scenario 2.2</i> | <i>Scenario 3</i> | <i>Scenario 3.2</i>     |
|---------------------------|-------------------------|------------------------|-------------------|---------------------|-------------------|-------------------------|
| <b>Redundancy</b>         | service                 | service                |                   |                     |                   | service                 |
| <b>Server Updates</b>     | no downtime, extra cost | no downtime extra cost | no                | no                  | yes               | no downtime, extra cost |
| <b>Server Maintenance</b> | simple                  | simple                 | very simple       | very simple         | average           | simple                  |
| <b>Launch Cost</b>        | €117752                 | €117752                | €102000           | €102000             | €6000             | €0                      |
| <b>Cost per Project</b>   | €525.6                  | €718.32                | €303.68           | €414.64             | €320              | €414.64                 |

After the organization of the information in the table it was done an analysis to verify if there was any scenario that was dominated by any other scenario. If that happened it would not be necessary to consider that scenario when applying the AHP methodology. A scenario A is said to dominate a scenario B if A is better than B in at least one criteria, without being worst than B in any of the remaining criteria. A quick analysis of Table 3 shows that none of the scenarios is dominated by other scenario.

After this analysis the information seemed to be ready to be used and analysed with the AHP methodology.

## 7.1.2 Applying AHP

To apply the process of decision aiding with AHP the Web application developed and described in Chapter 6 was used. Some of the steps of the process will now be described.

### 7.1.2.1 Filling-in the Criteria and the Scenarios

To use the information of the defined criteria with AHP it was necessary to first verify if all the criteria were independent of each other, since AHP does not support criteria dependence, and also define the hierarchic structure of the criteria, grouping the related criteria under a common criteria.

In terms of criteria dependency the criterion Redundancy and the criterion Load Balance share always the same value, no matter the scenario. Although they are different they are connected, because both are dependent of the server structure. This relation of dependence would influence the process, with alternatives getting double preference for the same characteristic of the solution, the server structure. Due to this it was decided to consider only one of these criteria in the application of AHP. It was decided to consider the criterion Load Balance. Besides this particular case all the other criteria were independent from each other.

Regarding the hierarchy among the criteria it was decided to arrange them in four groups, based on their similarity and their characteristics. These groups and the criteria that belong to them are the following:

- **CMST Related:** Hours of development, codebase type, Security, Built-in Applications, Ease of Use, Deployment, and Update Web Applications, Added

Value;

- **Hosting Related:** Portability, Scalability, Load Balance, Backup of Data and Applications, Backups Redundancy, Server Updates, and Server Maintenance;
- **Cost:** Launch Cost, and Cost per Project

After the hierarchy of the criteria was defined the information related with the criteria was added to the Web application, as it can be seen in Figure 14.

|    | Criterion Name          | Data Type  | Short Description                             | Parent Criterion |
|----|-------------------------|------------|---|------------------|
| 1  | CMST Related            | Parent     | Group of criteria related with the CMST ch    |                  |
| 2  | Hosting Related         | Parent     | Group of criteria related with the Hosting c  |                  |
| 3  | Cost                    | Parent     | Group of criteria related with the costs of t |                  |
| 4  | Added Value             | Integer    | How much added value is expected with th      | 1                |
| 5  | Hours of Development    | Integer    | How many hours are needed to do a proje       | 1                |
| 6  | Codebase type           | Binary     | Is it possible to have a single installation? | 1                |
| 7  | Security                | Integer    | Does the CMST provide security functionali    | 1                |
| 8  | Built-in Applications   | Integer    | What is the CMST value in terms of Built-ir   | 1                |
| 9  | Ease of Use             | Integer    | How well "equiped" is the CMST in terms o     | 1                |
| 10 | Deployment              | Complexity | How complex it is to deploy a Web Appli       | 1                |
| 11 | Update Web App.         | Complexity | How simple it is to update a Web App?         | 1                |
| 12 | Portability             | Easiness   | Is it easy to move one App to another host    | 2                |
| 13 | Scalability             | Integer    | How long in days does it take to scale up?    | 2                |
| 14 | Load Balance            | Integer    | per requests among different servers?[0,2]    | 2                |
| 15 | Backup of Data and Apps | Integer    | olution provides backup of the data? [0,2]    | 2                |
| 16 | Backups Redundancy      | Integer    | s by the hosting provider redundant? [0,2]    | 2                |
| 17 | Server Updates          | Integer    | Does server update cause downtime?[0,2]       | 2                |
| 18 | Server Maintenance      | Complexity | Is server maintenance complex?                | 2                |
| 19 | Launch Cost             | Monetary   | low much does it cost to launch a scenario    | 3                |
| 20 | Cost Per Project        | Monetary   | What is the yearly cost per project?          | 3                |

*Figure 14 Applying AHP - Criteria List*

After all criteria was added to the Web application the next step was to add the information of the different scenarios. The result of this step can be seen in Figure 15.

**Multi-Criteria Decision Analysis - AHP Methodology**

Now please fill the information about the scenarios.

Download XML Back Next Step Add More Remove Extra

|                              | Scenario 1  | Scenario 2    | Scenario3       | Scenario4         | Scenario5         | Scenario6      |
|------------------------------|-------------|---------------|-----------------|-------------------|-------------------|----------------|
| <b>Short Description</b>     | MOSS in AWS | MOSS in AWS.2 | Custom in Azure | Custom in Azure.2 | Umbraco in TerreM | Umbraco in AWS |
| <b>CMST Related:</b>         |             |               |                 |                   |                   |                |
| <b>Hours or Development</b>  | 1600        | 1600          | 320             | 320               | 400               | 400            |
| <b>Codebase type</b>         | Yes         | Yes           | No              | No                | No                | No             |
| <b>Security</b>              | 10          | 10            | 10              | 10                | 8                 | 8              |
| <b>Built-in Applications</b> | 8           | 8             | 6               | 6                 | 7                 | 7              |
| <b>Ease of Use</b>           | 9           | 9             | 10              | 10                | 8                 | 8              |
| <b>Deployment</b>            | Average     | Average       | Simple          | Simple            | Simple            | VerySimple     |
| <b>Update Web App.</b>       | Simple      | Simple        | Simple          | Simple            | Simple            | VerySimple     |
| <b>Added Value</b>           | 9           | 9             | 10              | 10                | 8                 | 8              |
| <b>Hosting Related:</b>      |             |               |                 |                   |                   |                |
| <b>Portability</b>           | VeryEasy    | VeryEasy      | Hard            | Hard              | VeryEasy          | VeryEasy       |

*Figure 15 Applying AHP - Scenarios Information*

In the Application the scenarios' name are created sequentially. So as can be seen in Figure 15 the scenarios are named differently from the names used in Chapter 5. To avoid confusion, especially when analysing the results, Table 4 Scenarios Denomination establishes the relation between the denomination used in Chapter 5 and the one present in the Web Application. In this chapter it will be used the denomination from the Web Application to facilitate the reference to the figures.

*Table 4 Scenarios Denomination*

| <i>Name in Chapter 5</i> | <i>Name in the Application</i> | <i>Description</i>  |
|--------------------------|--------------------------------|---|
| Scenario 1               | Scenario 1                     | MOSS 2007 in AWS  |
| Scenario 1.2             | Scenario 2                     | MOSS 2007 in AWS with bigger server structure               |
| Scenario 2               | Scenario 3                     | Custom CMST in Microsoft Azure                              |
| Scenario 2.1             | Scenario 4                     | Custom CMST in Microsoft Azure with bigger server structure |
| Scenario 3               | Scenario 5                     | Umbraco in TerreMark  |
| Scenario 3.2             | Scenario 6                     | Umbraco in AWS  |

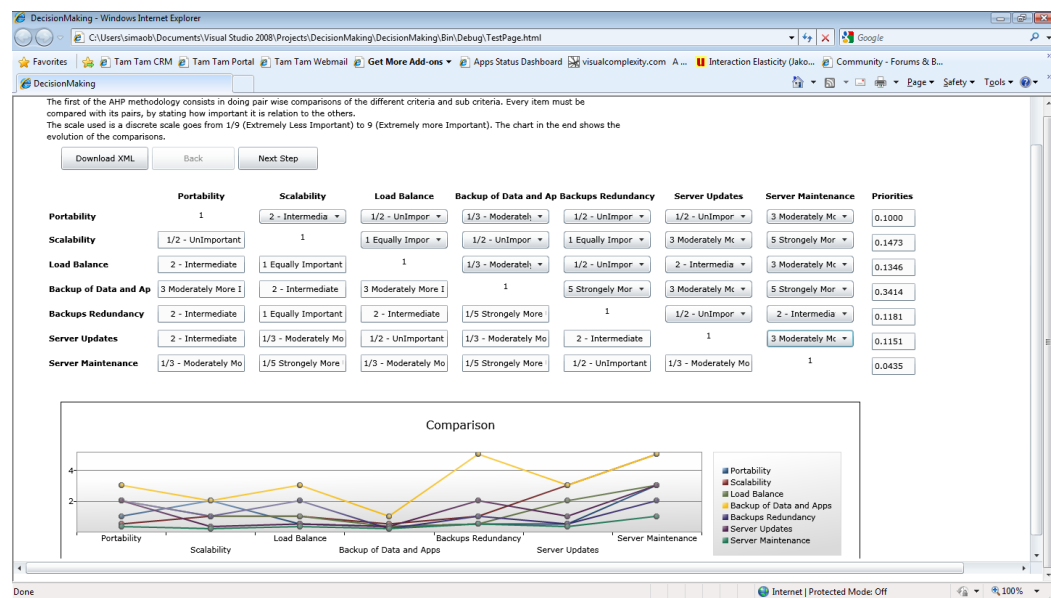
### 7.1.2.2 The Pair Wise Comparisons

After adding the information of the criteria and the scenarios it was time to start the pair wise comparisons towards to prioritize the alternatives. This was the longest part of the application of the AHP. The first comparisons were of the main criteria, then for each main criteria their sub

criteria were compared to obtain their priorities, and then it was time to compare all the scenarios regarding each criterion.

These comparisons were made through direct questioning of the Decision Makers (DMs). Together they defined their level of preference of one element over another for each pair wise comparison. This process was followed as carefully as possible to avoid inconsistencies. Inconsistencies normally fail to reflect the real preferences of the DMs. An example of an inconsistency is: When one defines that an element A is more important than an element B, and then defines that element B is more important than an element C, and finishing the comparison by defining that element A is less important than element C. Although inconsistency is inherent to a process like this where different judgements are made it should be avoided whenever possible. Otherwise, a high level of inconsistency will bias the results, and make them of low interest.

In Figure 16 one of the comparison screens can be seen. This one is the comparison of the sub criteria that are under the “Hosting Related” criteria.



*Figure 16 Applying AHP - Comparing Sub Criteria*

After the necessary comparisons were made the application calculated and presented the results of this process.

### 7.1.2.3 AHP Results

With the information provided the Web Application calculated the priority of each scenario. In the results page it is presented a ranking of the criteria, which can be seen in Figure 17. This ranking is based on the priorities calculate through the pair wise comparisons.

| Criteria                | Priority ▼ |
|-------------------------|------------|
| Backup of Data and Apps | 0.168      |
| Launch Cost             | 0.131      |
| Codebase type           | 0.101      |
| Scalability             | 0.073      |
| Load Balance            | 0.066      |
| Cost Per Project        | 0.065      |
| Backups Redundancy      | 0.058      |
| Server Updates          | 0.057      |
| Added Value             | 0.052      |
| Portability             | 0.049      |
| Security                | 0.046      |
| Ease of Use             | 0.044      |
| Hours of Development    | 0.022      |
| Server Maintenance      | 0.021      |
| Deployment              | 0.018      |
| Built-in Applications   | 0.017      |
| Update Web App.         | 0.011      |

*Figure 17 Applying AHP - Criteria Ranking*

In this ranking the criterion “Backup of Data and Applications” shows up as the one with highest priority, or preference. It is then followed by the “Launch Cost” criterion, and the “Codebase type” criterion. The ranking goes on listing all the criterion. In the last position is the criterion “Update Web Applications”.

Besides the global priorities of the criteria the Web application also calculates and displays the priorities of the scenarios. These priorities as well as the detailed priority of each scenario in relation to the different criteria, are shown in Figure 18, Figure 19, and Figure 20.

| Scenarios  | CMST Related         |               |              |                       |              |              |                 |              |
|------------|----------------------|---------------|--------------|-----------------------|--------------|--------------|-----------------|--------------|
|            | Hours of Development | Codebase type | Security     | Built-in Applications | Ease of Use  | Deployment   | Update Web App. | Added Value  |
| Scenario 1 | 0.001                | <b>0.036</b>  | <b>0.010</b> | <b>0.005</b>          | 0.007        | 0.001        | 0.002           | 0.008        |
| Scenario 2 | 0.001                | 0.036         | 0.010        | 0.005                 | 0.007        | 0.001        | 0.002           | 0.008        |
| Scenario 3 | <b>0.007</b>         | 0.007         | 0.010        | 0.001                 | <b>0.012</b> | 0.003        | 0.002           | <b>0.014</b> |
| Scenario 4 | 0.007                | 0.007         | 0.010        | 0.001                 | 0.012        | 0.003        | 0.002           | 0.014        |
| Scenario 5 | 0.003                | 0.007         | 0.003        | 0.002                 | 0.004        | 0.003        | 0.002           | 0.004        |
| Scenario 6 | 0.003                | 0.007         | 0.003        | 0.002                 | 0.004        | <b>0.006</b> | <b>0.003</b>    | 0.004        |
| Total      | 0.022                | 0.101         | 0.046        | 0.017                 | 0.044        | 0.018        | 0.011           | 0.052        |
| 0.311      |                      |               |              |                       |              |              |                 |              |

*Figure 18 Applying AHP - Results 1*

| Scenarios  | Hosting Related |              |              |                         |                    |                |                    |
|------------|-----------------|--------------|--------------|-------------------------|--------------------|----------------|--------------------|
|            | Portability     | Scalability  | Load Balance | Backup of Data and Apps | Backups Redundancy | Server Updates | Server Maintenance |
| Scenario 1 | <b>0.011</b>    | <b>0.014</b> | 0.011        | 0.014                   | 0.007              | 0.006          | 0.003              |
| Scenario 2 | 0.011           | 0.014        | <b>0.026</b> | 0.014                   | 0.007              | 0.006          | 0.003              |
| Scenario 3 | 0.002           | 0.014        | 0.004        | <b>0.042</b>            | <b>0.018</b>       | <b>0.018</b>   | <b>0.006</b>       |
| Scenario 4 | 0.002           | 0.014        | 0.011        | 0.042                   | 0.018              | 0.018          | 0.006              |
| Scenario 5 | 0.011           | 0.003        | 0.004        | 0.042                   | 0.003              | 0.003          | 0.001              |
| Scenario 6 | 0.011           | 0.014        | 0.011        | 0.014                   | 0.007              | 0.006          | 0.003              |
| Total      | 0.049           | 0.073        | 0.066        | 0.168                   | 0.058              | 0.057          | 0.021              |

0.492

Figure 19 Applying AHP - Results 2

| Scenarios  | Cost         |                  | Total         |
|------------|--------------|------------------|---------------|
|            | Launch Cost  | Cost Per Project |               |
| Scenario 1 | 0.004        | 0.006            | 0.1451        |
| Scenario 2 | 0.004        | 0.003            | 0.1580        |
| Scenario 3 | 0.009        | <b>0.018</b>     | <b>0.1864</b> |
| Scenario 4 | 0.009        | 0.010            | 0.1853        |
| Scenario 5 | 0.040        | 0.018            | 0.1530        |
| Scenario 6 | <b>0.063</b> | 0.010            | 0.1712        |
| Total      | 0.131        | 0.065            |               |

0.196      1.0000

Figure 20 Applying AHP - Results 3

In each column the highest value is represented in bold. These results point the scenario number 3, which is composed by a Custom CMST hosted in Azure Services Platform, as the one with higher priority.

Although this page is named the results page the AHP decision aiding process did not end here. The next step of the process was to analyse the results obtained.

### 7.1.3 Results Analysis

Before presenting the results analysis is important to remember that the AHP methodology is not a decision making methodology. AHP is a tool that is meant to help DMs support their decisions. The results obtained with the AHP methodology are not the path that the DMs should follow without question. These results are just a numerical way of representing the preferences of the DMs. The objective of the process is to make it easier for the DMs to look at the different criteria and the different scenarios and prioritize them. This was referred multiple times during the meeting where the Web application was applied. When using a methodology like this it is important that the DMs stay focused and understand that the methodology is not deciding for them, but instead it is helping them analyse the problem at hands.

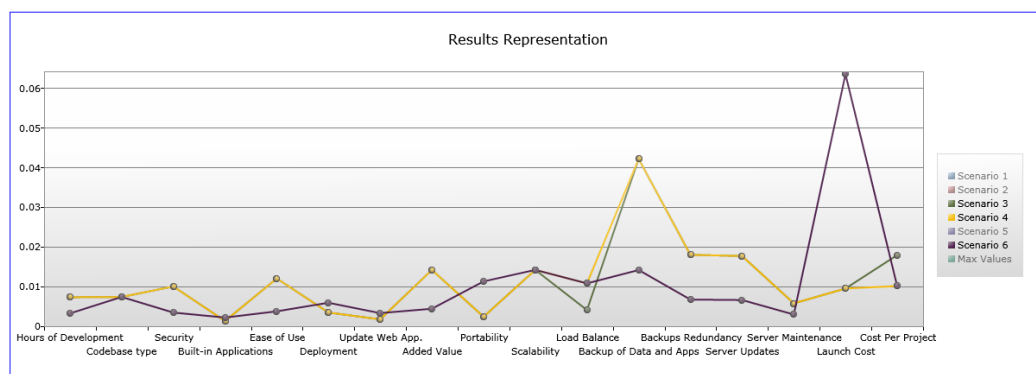
In the decision making meeting the analysis of the results was not very deep. The Web Application's sensitivity analysis was still a bit too simple and inconclusive to be used in the



meeting. Another option would have been to do the sensitivity analysis manually, but since the whole process is already too long by the end of it there was not much time left to do that. Nevertheless the results were looked into and discussed among the DMs.

The first thing that was discussed when was the criteria ranking (Figure 17). In the first position was placed the Backups of Data and Application, followed by Launch Cost, Codebase type, Scalability, and Load Balance, forming the five most preferred criteria, based on the DMs' inputs. Previously gathered information had shown that reduction in costs, the use of a single codebase CMST, and the possibilities provided by Cloud Computing (like scalability and load balance) were some of the reasons why Tam Tam wanted to analyse alternatives to their CMS Projects. So the DMs thought that it was just natural that these criteria ranked so high. In relation to the first place of the ranking it was seen with some surprise. Not that the DMs did not feel that it was an important criterion, to a company dedicated to Web development backups are essential for the safety of its customers' products. Nevertheless during the development of this project the backups was never one of the main focus of Tam Tam, maybe because in the actual situation they are already ensured. During the discussion two possible reasons were pointed out. Maybe the DMs really value this criterion even though it is not seen, in this case, as a major priority or a flaw that needs to be solved. Or during the questioning they misjudge some of the relations of the hosting related criteria.

Regarding the results of the AHP methodology, which are shown in Figure 18, Figure 19, and Figure 20, the DMs identified themselves with them. The scenario with the highest priority was Scenario 3 (presented as Scenario 2 in Chapter 5, as shown in Table 4) which is composed by the Custom CMST and Azure Services Platform as the hosting solution with the simplest server structure (1 Front end Web server and 1 Database server). This scenario had the final priority value of 0.1864 finishing with just a slightly higher priority than Scenario 4, which had a priority of 0.1853 and is a variation of Scenario 3 but with a larger server structure. In the third place was placed Scenario 6 with 0.1712, which is composed by Umbraco as the CMST and Amazon Web Services as the hosting provider.

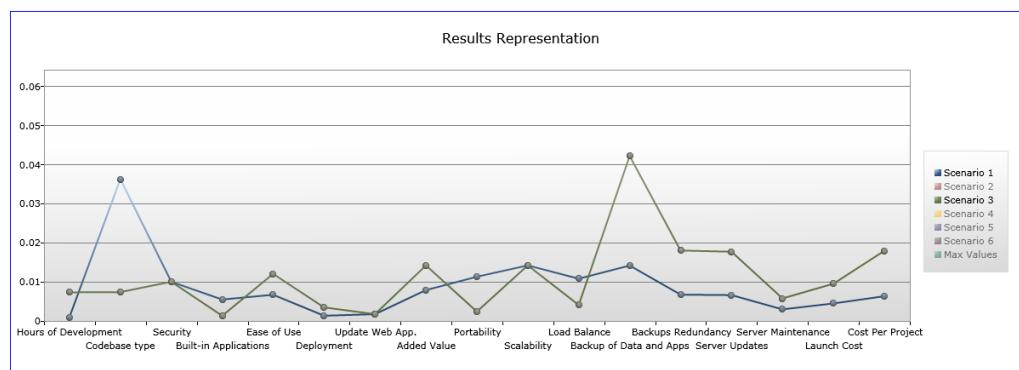


*Figure 21 Applying AHP - Compare Top 3 Scenarios*

In Figure 21 Applying AHP - Compare Top 3 Scenarios are represented the results of the 3 scenarios with highest priority. In this figure one can see that, as expected, Scenario 3 and Scenario 4 have very similar results and only differ in two criteria Load Balance and Cost Per Project. It is important to notice that these two criteria are placed sequentially in the criteria ranking. And as difference between the priorities of Scenario 3 and of Scenario 4 is very small,

this results might be sensitive to small changes in the criteria ranking or in the judgements of the DMs. Scenario 6 is is better classified than the other two in only three criteria Deployment, Portability and Launch Cost. In terms of Launch Cost, the second in the criteria ranking, the difference between Scenario 6 and the other two is very high. Nevertheless it is not enough to make it the scenario with the highest priority. One reason for this might be the fact that the other two scenarios also have a big difference towards Scenario 6 in terms of the number one ranking criteria.

These results was also discussed during the meeting. The DMs felt that the results reflected relatively well what they felt about the different scenarios. Scenario 3, and also Scenario 4 due to their similarities, were two of the alternatives which they liked the most. Also Scenario 6 was another alternative that they thought it could be good for Tam Tam.



*Figure 22 Applying AHP - Scenario 1 vs Scenario 3*

Looking again to Figure 20 one can see that Scenario 1 was the one with the lowest priority, 0.1451. Being MOSS 2007 one of the tools currently used by Tam Tam, the DMs were a bit surprised for it to be the lowest in terms of priority. They had though that the use of MOSS 2007 with a Cloud hosting solution would be a better alternative. Figure 22 shows a comparison of the results of Scenario 1 and Scenario 3. Scenario 1 has a better score in some of the criteria, but only in the CMST related criteria. In hosting related criteria and costs it scores mostly worst than Scenario 3.

After the small discussion regarding the results obtained the DMs felt that the use of AHP seemed to reflect in most of the cases their ideas and preferences. They seem to have enjoyed the use of the methodology, although they felt it to be a bit complex and a slow process. Regarding the Web application and its functionalities they seemed quite impressed, although some flaws have been identified. Nevertheless they felt it to be better than using an Excel sheet to implement the methodology.

In this meeting the Decision Makers have not made any decision towards the choice of the alternative. Although this was not part of the scope of this project it would have been interesting to see the results obtained with the AHP methodology being used in the company decision. Nevertheless this was just the first use of the application, and maybe after all the functionalities are completely finished there might be place for another use of the application and maybe then a actual decision might follow.

## **7.2 Summary**

This chapter describes the steps taken to execute the Web application developed with this projects main problem. Before the meeting with the decision makers the information was prepared to introduced in the application.

The execution of the application is described and the results are presented. The execution of the application ranked as the most preferred alternative the Scenario 3, which is, composed by a Custom CMST in Microsoft's Azure Services Platform, with a server structure of two servers (as described in this report in section 5.2.2.2 ). An analysis of the results was made during the meeting with the DMs and it is also described in this Chapter.



## Chapter 8

# Conclusions and Future Work

*There are two kinds of people, those who finish what they start and so on.*

Robert Byrne[141]

### 8.1 Conclusions

This project's main objective was to aid Tam Tam in finding alternatives to the way they develop their CMS projects, by doing desk research work in documenting the state of the art of Content Management Systems, Cloud Computing and Cloud Service Providers. As Tam Tam needs to define, analyse and assess some scenarios and these processes can often be time consuming, another major goal of the project was to find out how some way how this might be improved, in terms of time and expediency.

Those objectives are considered fulfilled.

New promising alternatives were defined by taking two important factors into account: the tool used to build the CMS application, and the hosting solution for those applications.

In terms of CMST the focus was on Open Source tools built with the .NET framework. While regarding hosting solutions, a new computing style, Cloud Computing, was the major alternative analysed. This new style is still under development but the currently available services and the services that are under development both seem to be promising.

Based on the research made together with the requirements and the constraints established by Tam Tam it was possible to define a set of six alternative scenarios which were then characterized and compared.

In order to help in the comparison between different alternatives, MCDA methodologies were used and were found to be a good way to guide Tam Tam through this process.

AHP was chosen from among different MCDA methodologies because it seemed to have the necessary characteristics to help with this decision process.

To facilitate the use of AHP and make the process faster, a Web application was built to implement it. And although not all the specified functionalities of the application were implemented its main functionalities were finished. With the main functionalities working, it was possible to execute the application and use the AHP methodology with Tam Tam's case. Although some results were obtained and discussed with the Decision Makers they have not yet decided which alternative they will use in their CMS Projects. Though it would have been interesting to see the contribution of the application being taken into consideration when making the decision, this was not part of the scope of this project.

Moreover, this type of decisions with a relevant impact in the company's strategy depend on several external factors that MCDA methodologies cannot account for and do not aim to. Nevertheless it was positive to see Tam Tam's Decision Makers using this methodology, welcoming it as a change in the way they normally make their decisions, normally more by following gut feeling and judgement, rather than using systematic ('rational') procedures. Thus, an helpful tool to support the choices that Decision Makers have to take was created and we believe that could help the company in saving of time.

Therefore, by defining, characterizing, analysing and comparing the different alternatives, and by choosing a MCDA methodology, implementing it in a Web Application, and introducing it to Tam Tam's Decision Makers, the author believes that the objectives of this project have been achieved.

## **8.2 Future Work**

After finishing this project some considerations can be made regarding possible follow up for the work developed. Cloud Computing is a type of computing under continuous development involving many companies trying to develop new and interesting services. It is important to stay thoroughly aware of all the changes that occur, just to keep competitive. Besides paying attention to what others are doing it could be also interesting to analyse the market and find a place for a product developed by Tam Tam specifically built for the Cloud that is new and innovative. Apart from the development of new products the simple leveraging of the hosting solutions provided by the Cloud could be expanded also to other products built by Tam Tam.

In terms of the research done a problem has been found that is shared by Cloud Computing and MCDA, although it is more complex for Cloud Computing. This problem is ambiguity, i.e. the more often than not lack of uniformity and consensus regarding the information. Completely different definitions for the same concept or for parts of the same concept, as well as different characteristics and details attributed to a concept are very common. This might be related with the lack of entities to define standards and concepts, but nevertheless it makes the understanding some important concepts somewhat complicated, making it also hard, and sometimes even impossible, to know what is the 'correct' definition of some concepts, if there is one. In terms of future, and although a bit unrelated with this project, it would be interesting to see researchers and IT experts work together to create ways to have the information available for consultation and as clear and unambiguous as possible. This would make bibliographic research a bit less complex and more rewarding.

Regarding the Web Application, the duration of the project was not enough to have time to do Usability Tests to the application. Nevertheless the DMs did not use the application by themselves and so it did not seem crucial to have Usability Tests at this point. These tests should be made in case the application is to be used by the DMs of Tam Tam or if it is to be integrated with other applications. In this case all the functionalities that are not yet finished should be completed before the Usability Tests are made.

Besides the functionalities specified in Section 6.2.1 some other functionalities could be added to improve the usefulness and the range of action of the application. Some possibilities are: to create a graphical representation of the problem's hierarchy; implementing other MCDA methodologies, that could work independently of the AHP or, if possible, in conjugation of AHP; and printing reports with a schematic and textual representation of the problem.





# References

1. Tam Tam. *Tam Tam*. Fullservice internetbureau gevestigd in Delft en Utrecht 2009; Available from: <http://www.tamtam.nl>.
2. Tam Tam. *Niews - Facts & Figures*. Over Tam Tam 2007 2009; Available from: <http://blogs.tamtam.nl/nieuws/2006/07/03/FactsFigures.aspx>.
3. Microsoft. *Partner Finder*. Tam Tam B.V. 2009; Available from: <http://www.microsoft.com/germany/aktionen/partnerfinden/default.aspx?partnerid=aaf b32cb-632d-4c52-b928-5e5702cb2ad0&tab=2>.
4. Microsoft. *Microsoft Office SharePoint Server*. Microsoft Office SharePoint Server 2007 2009; Available from: <http://www.microsoft.com/Sharepoint/default.aspx>.
5. Smartsite. *Homepage Smartsite Benelux*. Smartsite - Putting your content to work 2009; Available from: <http://www.smartsite.nl>.
6. Terremark. *Terremark Worldwide Inc*. Terremark 2009; Available from: <http://www.terremark.com/default.aspx>.
7. Collison, C. and G. Parcell, *Learning to Fly: Practical Knowledge Management from Leading and Learning Organizations*. 2004: Capstone Publishing LTD. 302.
8. Wikipedia. *Content Management*. Wikipedia, the free encyclopedia 2009 May 20; Available from: [http://en.wikipedia.org/wiki/Content\\_management](http://en.wikipedia.org/wiki/Content_management).
9. Wray, R. *Internet data heads for 500bn gigabytes*. guardian.co.uk - Business 2009 May 18; Available from: <http://www.guardian.co.uk/business/2009/may/18/digital-content-expansion>.
10. McKeever, S., *Understanding Web content management systems: evolution, lifecycle and market*. Industrial Management & Data Systems, 2003. **103**(9): p. 6.
11. WhatIs.com. *What is Webmaster?* WhatIs.com; Available from: [http://whatis.techtarget.com/definition/0\\_sid9\\_gci213349,00.html](http://whatis.techtarget.com/definition/0_sid9_gci213349,00.html).
12. Wikipedia. *Content Management System*. Wikipedia, the free encyclopedia 2008 May 18; Available from: [http://en.wikipedia.org/wiki/Content\\_management\\_system](http://en.wikipedia.org/wiki/Content_management_system).
13. Frederik. *Why CMS is not ECM(or WCM)?* Docbyte - Enterprise Content Management Experts 2008 November 1; Available from: <http://www.docbyte.com/blog/why-cms-not-ecm-or-wcm>.
14. ContentManager.eu.com. *So why all the fuss about 'content management'?* Enterprise Content Management Systems 2000-2008; Available from: <http://www.contentmanager.eu.com/index.htm>.
15. Enterprise Content Management System. *What is a Content Management System, or CMS?* Enterprise Content Management 2000-2008; Available from: <http://www.contentmanager.eu.com/cms.htm>.
16. Hoffman, R. *Content management tools*. Network Computing 2000 June 12; Available from: <http://www.networkcomputing.com/1111/1111buyers2.html>.
17. joviawebstudio. *Is a Content Management System Right for You?* Jovia Web Studio - Blog 2008 December 23; Available from: [http://www.joviawebstudio.com/jovia/blog/is\\_a\\_content\\_management\\_system\\_right\\_for\\_you/](http://www.joviawebstudio.com/jovia/blog/is_a_content_management_system_right_for_you/).

18. Enterprise Content Management System. *Reasons to purchase a Content Management System*. What are the benefits of Content 2000-2008; Available from: <http://www.contentmanager.eu.com/cmsbens.htm>.
19. aiim. *What is Web CMS (or WCM)?* aiim 2008; Available from: <http://www.aiim.org/What-is-Web-CMS-WCM-System-Content-Management.aspx>.
20. Wikipedia. *List of content management systems*. Wikipedia, the free encyclopedia 2009 June 3; Available from: [http://en.wikipedia.org/wiki/List\\_of\\_content\\_management\\_systems](http://en.wikipedia.org/wiki/List_of_content_management_systems).
21. OSCOM. *CMS Matrix*. Open Source Content Management 2002-2004; Available from: <http://oscom.org/matrix/>.
22. The CMS Matrix. *CMS Matrix*. The Content Management Comparison Tool 2009; Available from: <http://www.cmsmatrix.org/matrix>.
23. opensourceCMS. *ASP.NET CMS Demos*. opensourceCMS 2002-2009; Available from: <http://aspx.opensourcecms.com/>.
24. opensourceCMS. *PHP CMS Demos*. opensourceCMS 2002-2009; Available from: <http://php.opensourcecms.com/>.
25. System, E.C.M. *Open Source CMS Solutions*. Enterprise Content Management 2000-2008; Available from: <http://www.contentmanager.eu.com/opensourcevendors.htm>.
26. Kyrnin, J. *Content Management Tools*. About.com 2000 March 13; Available from: <http://webdesign.about.com/od/contentmanagement/a/aa031300c.htm>.
27. Kyrnin, J. *Choosing a CMS - What you need to know*. About.com 2004 May 17; Available from: <http://webdesign.about.com/cs/contentmgmt/a/aa051704.htm>.
28. Choo, G. *Buy, build, or rent that CMS?* TechRepublic 2002 August 20; Available from: [http://articles.techrepublic.com.com/5100-10878\\_11-1051947.html](http://articles.techrepublic.com.com/5100-10878_11-1051947.html).
29. CMSWatch. *The Web CMS Report 2009 from CMS Watch*. CMS Watch 2009; Available from: <http://www.cmswatch.com/CMS/Report/>.
30. idealware. *Comparing Open Source Content Management Systems: WordPress, Joomla, Drupal and Plone*. idealware 2009; Available from: [http://www.idealware.org/comparing\\_os\\_cms/](http://www.idealware.org/comparing_os_cms/).
31. Drupal. *About Drupal*. Drupal 2008 February 28; Available from: <http://drupal.org/about>.
32. GNU. *The GNU General Public License*. GNU Project - Free Software Foundation 2007 June 29; Available from: <http://www.gnu.org/copyleft/gpl.html>.
33. Drupal. *Community plumbing*. Drupal 2009 May 13; Available from: <http://drupal.org/>.
34. Joomla. *Joomla!* Joomla 2009 June 7; Available from: <http://www.joomla.org/>.
35. Mambo. *Mamboserver - Home*. mambo 2007; Available from: <http://www.mamboserver.com/>.
36. Joomla. *Joomla! Core Team*. Joomla! 2005-2009; Available from: <http://www.joomla.org/about-joomla/the-project/current-core-team.html>.
37. Joomla. *What is Joomla?* Joomla! 2005-2009; Available from: <http://www.joomla.org/about-joomla.html>.
38. Umbraco. *The Friendly CMS*. Umbraco 2009; Available from: <http://www.umbraco.org>.
39. The CMS Matrix. *Umbraco CMS*. The Content Management Comparison Tool 2009; Available from: <http://www.cmsmatrix.org/matrix/cms-matrix/umbraco-cms>.
40. Microsoft. *Office SharePoint Server 2007*. Microsoft TechNet 2009; Available from: <http://technet.microsoft.com/en-us/library/cc303422.aspx>.
41. Microsoft. *Microsoft Corporation*. Microsoft 2009; Available from: <http://www.microsoft.com/en-us/default.aspx>.
42. Microsoft. *Office Online Home Page*. Microsoft Office Online 2009; Available from: <http://office.microsoft.com/en-us/default.aspx>.
43. The CMS Matrix. *MOSS 2007*. The Content Management Comparison Tool 2007; Available from: <http://www.cmsmatrix.org/matrix/cms-matrix/moss-2007>.
44. Wikipedia. *Server (computing)*. Wikipedia, the free encyclopedia 2009 June 3; Available from: [http://en.wikipedia.org/wiki/Server\\_\(computing\)](http://en.wikipedia.org/wiki/Server_(computing)).

45. Wikipedia. *Web Server*. Wikipedia, the free encyclopedia 2009 June 2; Available from: [http://en.wikipedia.org/wiki/Web\\_Server](http://en.wikipedia.org/wiki/Web_Server).
46. W3C. *World Wide Web Consortium*. World Wide Web Consortium - Web Standards 2009 June 4; Available from: <http://www.w3.org/>.
47. W3C. *Web Services Glossary*. W3C 2004 February 11; Available from: <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
48. Wikipedia. *SOAP*. Wikipedia, the free encyclopedia 2009 May 31; Available from: <http://en.wikipedia.org/wiki/SOAP>.
49. Wikipedia. *Hypertext Transfer Protocol*. Wikipedia, the free encyclopedia 2009 June 3; Available from: [http://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol).
50. Network Working Group. *Hypertext Transfer Protocol - HTTP/1.1 - Request for Comments*. RFC2616 1999 June; Available from: <http://tools.ietf.org/html/rfc2616>.
51. Wikipedia. *Protocol (computing)*. Wikipedia, the free encyclopedia 2009 June 3; Available from: [http://en.wikipedia.org/wiki/Protocol\\_\(computing\)](http://en.wikipedia.org/wiki/Protocol_(computing)).
52. W3Schools. *Web Hosting Introduction*. w3schools 1999-2009; Available from: [http://www.w3schools.com/hosting/host\\_intro.asp](http://www.w3schools.com/hosting/host_intro.asp).
53. Group, N.W. *File Transfer Protocol (FTP) - Request for comments*. RFC 959 1985 October; Available from: <http://tools.ietf.org/html/rfc959>.
54. Wikipedia. *File Transfer Protocol*. Wikipedia, the free encyclopedia 2009 June 3; Available from: [http://en.wikipedia.org/wiki/File\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/File_Transfer_Protocol).
55. Wikipedia. *Web hosting service*. Wikipedia, the free encyclopedia 2009 May 16; Available from: [http://en.wikipedia.org/wiki/Web\\_hosting](http://en.wikipedia.org/wiki/Web_hosting).
56. Wikipedia. *Internet hosting service*. Wikipedia, the free encyclopedia 2009 March 27; Available from: [http://en.wikipedia.org/wiki/Internet\\_hosting\\_service](http://en.wikipedia.org/wiki/Internet_hosting_service).
57. telivo. *What is web hosting?* telivo - internet solutions 2003-2007; Available from: <http://www.telivo.com/faqc7q14.html>.
58. Wikipedia. *Bandwidth (computing)*. Wikipedia, the free encyclopedia 2009 May 28; Available from: [http://en.wikipedia.org/wiki/Bandwidth\\_\(computing\)](http://en.wikipedia.org/wiki/Bandwidth_(computing)).
59. W3Schools. *Web Hosting Types*. W3Schools 1999-2009; Available from: [http://www.w3schools.com/hosting/host\\_types.asp](http://www.w3schools.com/hosting/host_types.asp).
60. Web Host 2 Go. *Types of Web Hosting*. Web Host 2 Go 2002-2009; Available from: <http://www.webhost2go.com/web-hosting/types.html>.
61. Goldberg, D. *Different Types of Web Hosting*. Ask Web Hosting 2000-2009; Available from: [http://www.askwebhosting.com/article/105/Different\\_Types\\_of\\_Web\\_Hosting.html](http://www.askwebhosting.com/article/105/Different_Types_of_Web_Hosting.html).
62. Hosting 411. *Types of Web Hosting and Web Hosting Providers*. Hosting 411; Available from: <http://www.hosting411.info/>.
63. Wikipedia. *Web hosting service - Types of hosting*. Wikipedia, the free encyclopedia 2009 May 16; Available from: [http://en.wikipedia.org/wiki/Web\\_hosting\\_service#Types\\_of\\_hosting](http://en.wikipedia.org/wiki/Web_hosting_service#Types_of_hosting).
64. Web Hosting GEEKS. *Different Types of Web Hosting*. Web Hosting GEEKS 2009 June 4; Available from: [http://webhostinggeeks.com/articles/Different\\_Types\\_of\\_Web\\_Hosting.html](http://webhostinggeeks.com/articles/Different_Types_of_Web_Hosting.html).
65. Wikipedia. *Virtual Private Server*. Wikipedia, the free encyclopedia 2009 May 23; Available from: [http://en.wikipedia.org/wiki/Virtual\\_private\\_server](http://en.wikipedia.org/wiki/Virtual_private_server).
66. Web Host 2 Go. *Virtual Private Servers Hosting Primer*. Web Hosting Basics and Reviews 2002-2006; Available from: <http://www.webhost2go.com/web-hosting/vps.html>.
67. Web Host 2 Go. *Dedicated Web Hosting*. Web Host 2 Go 2002/2006; Available from: <http://www.webhost2go.com/web-hosting/dedicated.html>.
68. Web Host 2 Go. *Reseller Web Hosting*. Web Host 2 Go 2002-2006; Available from: <http://www.webhost2go.com/web-hosting/reseller.html>.
69. Web Host 2 Go. *Co-Located Web Hosting*. Web Host 2 Go 2002-2006; Available from: <http://www.webhost2go.com/web-hosting/colocation.html>.

70. Geelan, J. *Twenty-One Experts Define Cloud Computing*. Cloud Computing Journal 2009 January 24; Available from: <http://cloudcomputing.sys-con.com/node/612375>.
71. Youseff, L., M. Butrico, and D. Da Silva. *Toward a Unified Ontology of Cloud Computing*. in *Grid Computing Environments Workshop, 2008*. 2008. Austin, TX: Univ. of California, Santa Barbara.
72. Stanzone, D., *So, What Actually is a Cloud?* 2009, Arizona State University: Arizona. p. 75.
73. Brown, M.M. *What cloud computing isn't*. Computerworld - Blogs 2008 October 7; Available from: [http://blogs.computerworld.com/what\\_cloud\\_computing\\_isnt](http://blogs.computerworld.com/what_cloud_computing_isnt).
74. Vaquero, L.M., et al., *A break in the clouds: towards a cloud definition*. SIGCOMM Comput. Commun. Rev., 2009. **39**(1): p. 50-55.
75. Schwartz, E. *The dangers of cloud computing*. InfoWorld - Analysis 2008 July 7; Available from: [http://www.infoworld.com/article/08/07/07/28NF-cloud-computing-security\\_1.html](http://www.infoworld.com/article/08/07/07/28NF-cloud-computing-security_1.html).
76. McAllister, N. *Cloud computing on whose terms?* 2008 July 31; Available from: [http://weblog.infoworld.com/fatalexception/archives/2008/07/cloud\\_computing.html](http://weblog.infoworld.com/fatalexception/archives/2008/07/cloud_computing.html).
77. Soltero, J. *The Cloud Dilemma for Developers*. Blogging Hyperic 2009 February 3; Available from: <http://www.hyperic.com/blog/cloud-dilemma-for-developers/>.
78. Dignan, L. *The race to 99.999 percent uptime: 3Tera ups the cloud SLA ante*. ZDNet.com - Between the Lines 2009 March 19; Available from: <http://blogs.zdnet.com/BTL/?p=14885>.
79. Gohring, N. *Amazon Web Services has another outage*. InfoWorld - Cloud Computing 2008 April 7; Available from: <http://www.infoworld.com/d/cloud-computing/amazon-web-services-has-another-outage-093>.
80. Gohring, N. *Amazon's S3 down for several hours*. InfoWorld - Platforms 2008 February 15; Available from: <http://www.infoworld.com/t/platforms/amazons-s3-down-several-hours-467>.
81. Rogers, G. *Google goes down, questions about cloud computing arise*. ZDNet.com - Googling Google 2009 May 16; Available from: <http://blogs.zdnet.com/Google/?p=1422>.
82. Dawson, C. *Yes, the cloud is still OK, even after Google's "glitch"*. ZDNET.com - Education IT 2009 May 14; Available from: <http://education.zdnet.com/?p=2551>.
83. Vijayan, J. *Report cites potential privacy gotchas in cloud computing*. Computerworld - Networking & Internet 2009 February 25; Available from: [http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=internet\\_applications&articleId=9128636&taxonomyId=168](http://www.computerworld.com/action/article.do?command=viewArticleBasic&taxonomyName=internet_applications&articleId=9128636&taxonomyId=168).
84. Brown, M.M. *Cloud computing interoperability*. Computerworld Blogs 2009 February 9; Available from: [http://blogs.computerworld.com/cloud\\_computing\\_interoperability](http://blogs.computerworld.com/cloud_computing_interoperability).
85. Urquhart, J. *Exploring cloud interoperability, part 1*. CNET News - Wisdom of Clouds 2009 May 2; Available from: [http://news.cnet.com/8301-19413\\_3-10231290-240.html](http://news.cnet.com/8301-19413_3-10231290-240.html).
86. Urquhart, J. *Exploring cloud interoperability, part 2*. CNET News - The Wisdom of Clouds 2009 May 7; Available from: [http://news.cnet.com/8301-19413\\_3-10235492-240.html](http://news.cnet.com/8301-19413_3-10235492-240.html).
87. Urquhart, J. *Exploring cloud interoperability, part 3*. CNET News - The Wisdom of Clouds 2009 May 16; Available from: [http://news.cnet.com/8301-19413\\_3-10241138-240.html](http://news.cnet.com/8301-19413_3-10241138-240.html).
88. Brandel, M. *Cloud computing: Don't get caught without an exit strategy*. InfoWorld News 2009 March 3; Available from: [http://www.infoworld.com/article/09/03/03/Cloud\\_computing\\_Dont\\_get\\_caught\\_witho\\_ut\\_an\\_exit\\_strategy\\_1.html](http://www.infoworld.com/article/09/03/03/Cloud_computing_Dont_get_caught_witho_ut_an_exit_strategy_1.html).
89. Rodrigues, S. *Cloud computing is overhyped*. InfoWorld - Open Source 2008 September 30; Available from: <http://www.infoworld.com/d/open-source/cloud-computing-overhyped-113?page=0,1>.
90. Veitch, M. *Poll: Cloud Computing not seen as IT answer to downturn*. ComputerworldUK - Community 2009 March 16; Available from:



- <http://www.computerworlduk.com/community/blogs/index.cfm?blogid=21&entryid=1991>.
91. Computerworld. *News, Education & Headlines - Security, Storage & Network - Information on Hardware, Software for Laptops & Desktops*. Computerworld 1994-2009; Available from: <http://www.computerworld.com>.
  92. Stedman, C. *Cloud computing not fully enterprise-ready, IT execs say*. Computerworld - Networking & Internet 2009 March 2; Available from: [http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9128840&intsrc=news\\_ts\\_head](http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9128840&intsrc=news_ts_head).
  93. Hall, M.E. *Cloud computing safer than on-premises*. Computerworld - Blogs 2009 March 9; Available from: [http://blogs.computerworld.com/cloud\\_computing\\_safer\\_than\\_on\\_premises](http://blogs.computerworld.com/cloud_computing_safer_than_on_premises).
  94. Havenstein, H. *Cloud computing takes hold despite privacy fears*. Computerworld - Networking & Internet 2008 September 15; Available from: <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9114730>.
  95. Diaz, S. *Google outage isn't reason enough to avoid a cloud strategy*. ZDNet.com - Between the Lines 2009 May 14; Available from: <http://blogs.zdnet.com/BTL/?p=18092>.
  96. Vaughan-Nichols, S.J. *After Googlefail, will you trust online apps?* Computerworld Blogs 2009 May 14; Available from: [http://blogs.computerworld.com/after\\_googlefail\\_will\\_you\\_trust\\_online\\_apps](http://blogs.computerworld.com/after_googlefail_will_you_trust_online_apps).
  97. Stevens, H. and C. Pettey. *Gartner Says Cloud Computing Will Be As Influential As E-business*. Gartner - Newsroom 2008 June 26; Available from: <http://www.gartner.com/it/page.jsp?id=707508>.
  98. Berger, I.W. *Cloud Computing Promise & Reality*. AlwaysOn 2008 July 14; Available from: <http://alwayson.goingon.com/permalink/post/28058>.
  99. *The Open Cloud Manifesto*. 2009: [www.opencloudmanifesto.org](http://www.opencloudmanifesto.org).
  100. Kost, J.M., S. 946, *the Information Technology Management Reform Act of 1995: hearing before the Subcommittee on Oversight of Government Management and the District of Columbia of the Committee on Governmental Affairs*. 1995: District of Columbia.
  101. Figueira, J., S. Greco, and M. Ehrgott, *Multiple criteria decision analysis: state of the art surveys*. illustrated ed. 2005: Springer. 1045.
  102. Bana E Costa, C.A., T.J. Stewart, and J.-C. Vansnick, *Multicriteria decision analysis: Some thoughts based on the tutorial and discussion sessions of the ESIGMA meetings*. European Journal of Operational Research, 1997. **99**(1): p. 28-37.
  103. Roy, B., D. Bouyssou, and P. Vincke, *Aiding decisions with multiple criteria: essays in honor of Bernard Roy*. illustrated ed. 2001: Springer. 558.
  104. Lootsma, F.A., *Multi-criteria decision analysis via ratio and difference judgement*. illustrated ed. 1999: Springer. 285.
  105. Pietersen, K., *Multiple criteria decision analysis (MCDA) : A tool to support sustainable management of groundwater resources in South Africa*. Water S.A., 2006. **32**(2): p. 10.
  106. Belton, V. and T.J. Stewart, *Multiple Criteria Decision Analysis: an integrated approach*. 2, illustrated ed. 2001: Springer. 372.
  107. Roy, B. and M.R. McCord, *Multicriteria methodology for decision aiding*. illustrated ed. 1996: Springer. 292.
  108. UK Department for Transport, L.G.a.t.R.D., *DTLR manual on Multi Criteria Analysis*. 2001. p. 145.
  109. Belton, V. and M. Elder, *Integrated MCDA : A simulation case study*. Lecture notes in economics and mathematical systems, 1998. **465**: p. 12.
  110. Herath, G. and T. Prato, *Using multi-criteria decision analysis in natural resource management*. illustrated ed. 2006: Ashgate Publishing, Ltd. 239.

111. Roy, B. and D. Vanderpooten, *The European school of MCDA: Emergence, basic features and current works*. Journal of Multi-Criteria Decision Analysis, 1996. **5**(1): p. 22-38.
112. Lootsma, F.A. *Multi-Criteria Decision Analysis and Multi-Objective Optimization at the Delft University of Technology, Faculty of Mathematics and Informatics*. Available from: <http://www.inescc.pt/~ewgmcda/Lootsma.html>.
113. Vassilev, V., K. Genova, and M. Vassileva, *Brief Survey of Multi-criteria Decision Making Methods and Software Systems*. Cybernetics and Information Technologies, 2005. **5**(1).
114. Design Decisions Wiki. *Multiattribute utility theory*. design decisions wiki 2009 March 17; Available from: [http://ddl.me.cmu.edu/ddwiki/index.php/Multiattribute\\_utility\\_theory](http://ddl.me.cmu.edu/ddwiki/index.php/Multiattribute_utility_theory).
115. Kangas, J. and A. Kangas, *Multiple criteria decision support in forest management--the approach, methods applied, and experiences gained*. Forest Ecology and Management, 2005. **207**(1-2): p. 133-143.
116. von Winterfeldt, D. and W. Edwards, *Decision Analysis and Behavioral Research*. 1986, Cambridge: Cambridge University Press.
117. Saaty, T.L., *How to Make a Decision: The Analytic Hierarchy Process*. INTERFACES, 1994. **24**(6): p. 19-43.
118. Doumpos, M. and C. Zopounidis, *Multicriteria decision aid classification methods*. illustrated ed. 2002: Springer. 252.
119. Roy, B., *The outranking approach and the foundations of electre methods*. Theory and Decision, 1991. **31**(1): p. 49-73.
120. About.Com. *Martin Luther King Jr. Quotes*. About.com: Quotations 2009; Available from: <http://quotations.about.com/od/stillmorefamouspeople/a/martinluther1.htm>.
121. InternetWorldStats. *European Union Internet usage and Population Stats*. Internet World Stats 2008 December; Available from: <http://www.internetworldstats.com/europa.htm#nl>.
122. QuoteDB. *Henry Kissinger*. QuoteDB; Available from: <http://www.quotedb.com/quotes/1468>.
123. Cuyahoga. *Cuyahoga. A .NET Web Site Framework* 2009 June 24; Available from: <http://www.cuyahoga-project.org>.
124. DotNetNuke. *The leading Open Source Web Content Management Framework for ASP.NET*. DotNetNuke 2002-2009; Available from: <http://www.dotnetnuke.com/>.
125. Wordpress. *Blog Tool and Publishing Platform*. WordPress.ORG 2009; Available from: <http://www.wordpress.org>.
126. Johnston, J.S. *Microsoft Says Cloud Offering Will be Price-Competitive*. IT Channel Planet 2009 February 12; Available from: [http://www.itchannelplanet.com/business\\_news/article.php/3802616/Microsoft+Says+Cloud+Offering+Will+Be+PriceCompeti.htm](http://www.itchannelplanet.com/business_news/article.php/3802616/Microsoft+Says+Cloud+Offering+Will+Be+PriceCompeti.htm).
127. Microsoft. *Services - Windows Azure*. Azure Services Platform 2008; Available from: <http://www.microsoft.com/azure/windowsazure.mspix>.
128. Microsoft. *Services - SQL Services*. Azure Services Platform 2008; Available from: <http://www.microsoft.com/azure/sql.mspix>.
129. Amazon Web Services. *Running Windows Server and SQL Server*. Amazon Elastic Compute Cloud (EC2) 2009; Available from: <http://aws.amazon.com/windows/>.
130. QuoteWorld.org. *Midori Koto*. Quote World 2009; Available from: <http://www quoteworld.org/quotes/7929>.
131. Smith, J.E. and D.v. Winterfeldt, *Decision Analysis in Management Science*. Management Science, 2004. **50**(5): p. 14.
132. Warren, L. and D.S.a.T.O. (Australia). *Uncertainties in the analytic hierarchy process / Lewis Warren*. 2004, Edinburgh, S. Aust. :. 22 p. :.
133. Perez, J., *Some comments on Saaty's AHP*. Manage. Sci., 1995. **41**(6): p. 1091-1095.

134. Wang, Y.-M. and T.M.S. Elhag, *An approach to avoiding rank reversal in AHP*. Decis. Support Syst., 2006. **42**(3): p. 1474-1480.
135. Zanakis, S.H., et al., *Multi-attribute decision making: A simulation comparison of select methods*. European Journal of Operational Research, 1998. **107**(3): p. 507-529.
136. Byun, D.-H., *The AHP approach for selecting an automobile purchase model*. Information & Management, 2001. **38**(5): p. 289-297.
137. Ayag, Z. and R.G. Ozdemir, *A combined fuzzy AHP-goal programming approach to assembly-line selection*. J. Intell. Fuzzy Syst., 2007. **18**(4): p. 345-362.
138. Takahagi. *AHP (Analytic Hierarchy Process) Calculation Software by CGI*. Available from: <http://www.isc.senshu-u.ac.jp/~thc0456/EAHP/AHPweb.html>.
139. Microsoft. *The Official Microsoft Silverlight Site*. Microsoft Silverlight 2009; Available from: <http://www.silverlight.net>.
140. Pasteur, L., *Revue Scientifique*. 1871: Paris.
141. Wikipedia. *Robert Byrne*. Wikipedia, the free encyclopedia 2009 June 6; Available from: [http://en.wikipedia.org/wiki/Robert\\_Byrne](http://en.wikipedia.org/wiki/Robert_Byrne).





# Index

- .NET, 22, 34, 38, 39, 40, 49, 59, 81
- AHP, 28, 29, 30, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 69, 71, 72, 73, 74, 75, 76, 77, 78, 81, 82, 83
- AWS, 21, 44, 45, 46, 52, 73
- Cloud Computing, i, iii, 3, 4, 5, 14, 16, 17, 18, 19, 20, 21, 23, 37, 40, 77, 81, 82
- CMS, i, iii, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 17, 23, 33, 34, 35, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47, 53, 56, 67, 77, 81, 82
- CMST, i, 1, 3, 4, 10, 39, 41, 42, 43, 44, 46, 47, 48, 49, 50, 51, 71, 73, 76, 77, 78, 79, 81
- DM, 26, 55
- ELECTRE, 28, 30, 31, 55
- MCDA, 4, 25, 26, 27, 28, 29, 30, 31, 35, 55, 56, 58, 67, 69, 81, 82, 83
- Microsoft, 2, 3, 21, 22, 33, 38, 39, 40, 43, 46, 47, 48, 56, 59, 65, 73, 79
- Scenarios, 4, 37, 39, 41, 50, 61, 62, 70, 71, 73, 77
- SMART, 29, 55, 56